

DESIGN, CHARACTERIZATION, AND CONTROLS FOR A SURGEON-INTERACTIVE  
ROBOTIC IMAGING SYSTEM

By

TIMOTHY Z. ELMORE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2013

© 2013 Tim Elmore

I dedicate this dissertation to my family, friends, and coworkers, for without their support such work would not have been possible.

## ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Scott Banks, for his endless wisdom, support, and dedication towards this research. He was without a doubt the most vital part of this research. I also thank my committee, Dr. B.J. Fregly, Dr. Carl Crane, and Dr. Frank Bova for their guidance and input. Finally, I would like to thank my fellow lab members Ira Hill, Nick Dunbar, Jared Jones, David Walker, and Brian Park for their support, intellectual contributions, and constant encouragement.

## TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS .....	4
LIST OF TABLES.....	7
LIST OF FIGURES .....	8
ABSTRACT .....	10
CHAPTER	
1 INTRODUCTION .....	11
Background.....	11
The Need for Image Guidance.....	11
History of Image Guided Surgery .....	11
Current State of Image Guided Surgery .....	12
A New Approach – Two Robotic Manipulators.....	14
Manipulator Design .....	15
Trajectory Planning .....	16
Reconstruction Sensitivity.....	16
2 MANIPULATOR DESIGN.....	18
The Goal .....	18
Introduction.....	18
Methods .....	19
Implementation and Experimental Assessment.....	24
Experimental Setup .....	27
Results.....	30
Discussion.....	34
Conclusion .....	35
3 MANIPULATOR CONTROL AND COORDINATION .....	36
The Goal .....	36
Introduction.....	36
Methods .....	38
Pre-planned Control.....	38
Haptic Control .....	41
Haptic Study .....	44
Protection of Human Subjects .....	46
Results.....	47
Discussion.....	49
Conclusion .....	49

4	TRAJECTORY PLANNING .....	51
	The Goal .....	51
	Description of Reconstruction Algorithm .....	51
	Problem Description .....	52
	Branch and Bound Algorithm.....	55
	Conclusion .....	60
5	CT RECONSTRUCTION SENSITIVITY ANALYSIS .....	62
	The Goal .....	62
	Introduction.....	62
	Methods .....	63
	Results.....	66
	Discussion.....	70
	Conclusion .....	71
6	CONCLUSION.....	73
	APPENDIX: BACK PROJECTION CONFIGURATION .....	74
	LIST OF REFERENCES .....	76
	BIOGRAPHICAL SKETCH.....	81

## LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 Datasets collected with the experimental system.....	28
2-2 Changes to <i>BASE</i> , <i>UTRFAROARM</i> T as a result of calibration .....	31
2-3 Original D-H parameters for the UTR .....	31
2-4 Modifications to the D-H parameters for the UTR after calibration, unloaded.....	31
2-5 Modifications to the D-H parameters for the UTR after calibration, 5kg load.....	31
2-6 Modifications to the D-H parameters for the UTR after calibration, 28kg load.....	32
2-7 Error of calibrated model .....	34
3-1 Settling times for a typical 150mm move.....	47
3-2 Time and force data for the nine position study .....	48
4-1 Time and path length for brute force .....	54
4-2 Time and number of unique paths for branch and bound.....	57
4-3 The UTR's dynamic constraints for path planning purposes.....	57
4-4 Time to complete both the random and optimally sorted paths .....	59
5-1 Error types for cases evaluated.....	64
5-2 RMS error and correlation coefficient for cases 1-5 .....	67

## LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1 The human spine, with vertebrae labeled .....	12
1-2 Current intraoperative imaging solutions A.) the C-arm B.) O-arm .....	13
1-3 Siemens Artis Zeego .....	13
1-4 Proposed configuration with X-ray source attached to under table robot (UTR) and x-ray image detector connected to Mitsubishi PA-10-6CE robotic arm .....	14
2-1 SolidWorks model of UTR.....	20
2-2 Available torque as a function of velocity .....	22
2-3 Joint 2's position, velocity, acceleration, and torque as a function of time .....	22
2-4 Labeled joint axes of the UTR .....	23
2-5 A set of 50 unique positions .....	26
2-6 Experimental setup showing PA10 and UTR.....	28
2-7 Solidworks FEA model of typical UTR arm loading during acceleration.....	30
2-8 Position error for unloaded case. Red points are not included in the optimization, and are present to verify the calibration. All points are included in reported error metrics.....	32
2-9 Position error for 5kg load case. Red points are not included in the optimization, and are present to verify the calibration.....	33
2-10 Position error for 28kg full payload case. Red points are not included in the optimization, and are present to verify the calibration. ....	33
3-1 Block diagram of PA10 control.....	38
3-2 Spine with light sensor positions noted. ....	44
3-3 Actual apparatus with light sensors. ....	45
3-4 Controller performance for a typical 150mm move .....	47
3-5 Controller performance for a near-collision.....	48
4-1 An example trajectory.....	53
4-2 An example tree.....	55

4-3 Branch and bound pseudo-code .....	56
4-3 First 10 rows and columns of the cost matrix C .....	58
5-1 MTF as a function of spatial frequency for cases 1-6.....	67
5-2 Modified Shepp-Logan phantom with white horizontal line indicating location of 1-D intensity plot.....	67
5-3 Intensity plots for a cross section of the modified Shepp-Logan phantom reconstruction....	68
5-4 Reconstructions of cranial CT scan .....	69

Abstract of Dissertation Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Doctor of Philosophy

DESIGN, CHARACTERIZATION, AND CONTROLS FOR A SURGEON-INTERACTIVE  
ROBOTIC IMAGING SYSTEM

By

Timothy Z. Elmore

December 2013

Chair: Scott Banks  
Cochair: B.J. Fregly  
Major: Mechanical Engineering

A surgeon-interactive robotic imaging system can provide improvements in the field of surgical imaging such as reduction of X-ray dose, improved reconstruction quality, and reduced overall size and bulk, improving surgeon access. However, using robotic manipulators to perform surgical imaging is not trivial. A new robotic manipulator must be designed, and there are dynamic and control challenges that must be addressed. Finally, there are path planning challenges, finding the optimum trajectory through a cloud of desired views, while avoiding collisions and meeting other safety constraints. Design of a novel manipulator and static identification of the kinematic and stiffness properties of that manipulator are performed. Coordination of a serial-link manipulator and a prismatic/revolute manipulator is demonstrated. Optimal trajectory planning is accomplished using a variation of methods developed for the classic traveling salesman problem. The prototype system demonstrates feasibility for all aspects of a surgeon-interactive robotic imaging system to provide a suite of novel and useful capabilities for image-guided surgical procedures.

## CHAPTER 1 INTRODUCTION

### **Background**

Back pain costs more than \$100 billion annually in the US. Over 650,000 surgical procedures are performed annually, with costs exceeding \$20 billion [1]. Spine surgery is benefited by some form of navigation, or image guided surgery (IGS) in order to properly place hardware in particular vertebrae. As of the late 1990s, four approaches were used: live fluoroscopic guidance, pre-surgical CT scans, virtual fluoroscopy, and 2D-3D registration. Although these technologies resulted in significant patient benefits, their success was limited by inaccurate registrations, increased surgeon radiation exposure, and/or the unrealistic assumption of rigid alignment between the CT scan and the operating room.

### **The Need for Image Guidance**

The success of spine surgery is dependent upon the placement of the hardware. Studies have been performed on the accuracy requirements for pedicle screw placement [2]. It was reported that maximum permissible translational/rotational error tolerances ranged from 0.0mm/0.0° at T5 to 3.8mm/12.7° at L5. The human spine is shown in Figure 1-1. Therefore, with zero permissible error of some vertebrae, improvements in navigation during spine surgery promise to improve patient outcomes. Furthermore, we see that there are other areas of improvement, such as patient radiological dose, equipment size, and equipment cost.

### **History of Image Guided Surgery**

Over the past 10 years, a new technology has emerged that has expanded utilization of IGS. Intraoperative cone beam computed tomography (CBCT) has provided significant advances, reducing the drawbacks associated with the aforementioned approaches, while providing useful views and measurements to the surgeon. However, one of the biggest

drawbacks to the current CBCT systems is the bulk of the machine, and its interference with the surgeon's approach to the surgical field. Additionally, current technologies have limited mechanical degrees of freedom, resulting in increased dose to the patient to provide the same quality reconstruction.

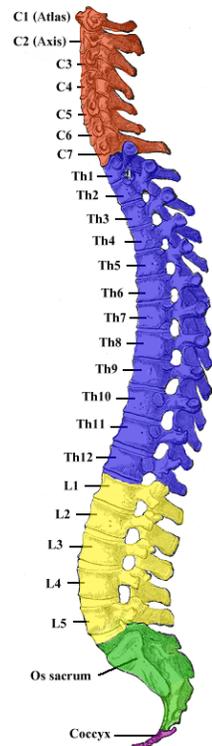


Figure 1-1. The human spine, with vertebrae labeled [3]

### **Current State of Image Guided Surgery**

Current navigation methods have limitations, namely cost, limited imaging capabilities, and bulk, resulting in crowding an already overcrowded operating room environment. The popular imaging tools for the operating room currently are the C-Arm and O-arm, as shown in Figure 1-2.

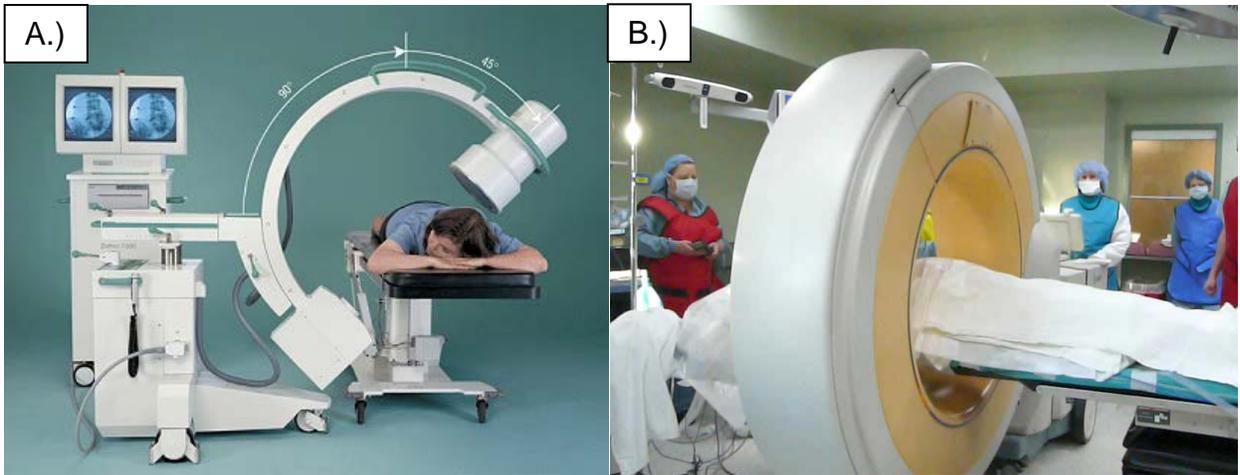


Figure 1-2. Current intraoperative imaging solutions A.) the C-arm [4] B.) O-arm [taken by investigator]

A robotic imaging system can overcome these limitations, but development of such a system presents many challenges that have not been addressed. The Siemens Artis Zeego, a new imaging system, attempts to address these issues by mounting a conventional C-arm on a robot arm, but the required arm is 2-3 times bulkier than existing O-arm technology.



Figure 1-3. Siemens Artis Zeego [5]

Additionally, the available views are still limited, as the SID (source to image distance) is fixed, and the alignment of the source and sensor are also fixed. Betz et al. [6] suggested in 2002 there

might be advantages to having dual-robot imaging system, but no surgical imaging platforms to date have put the x-ray source and x-ray detector on separate robotic manipulators.

### **A New Approach – Two Robotic Manipulators**

The solution shown in Figure 1-4 seeks to overcome all existing limitations without compromise. By attaching the x-ray source and detector directly to robots, the bulk of the manipulators can be greatly reduced, while allowing 6-DOF control over the image detector, and 5-DOF control over the x-ray source.

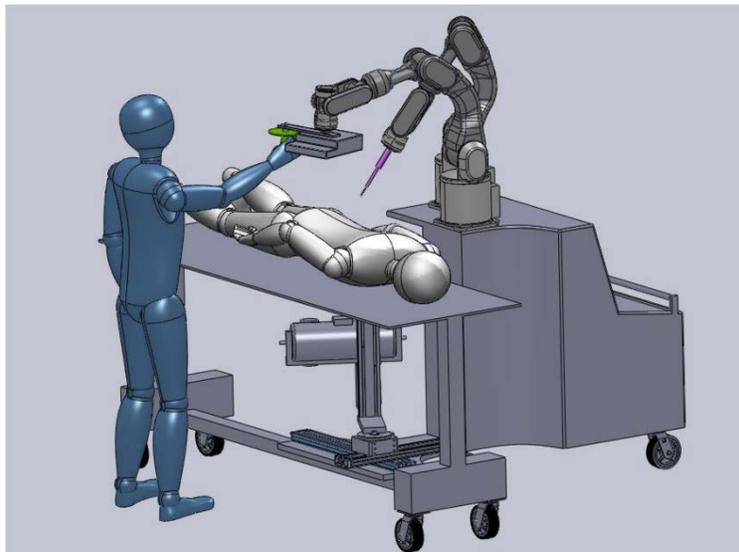


Figure 1-4. Proposed configuration with X-ray source attached to under table robot (UTR) and x-ray image detector connected to Mitsubishi PA-10-6CE robotic arm

Until recently, CT reconstruction algorithms were limited by computational complexity to methods requiring equal spacing of views, typically rotating about a single fixed axis. Recent work has shown a constellation of views to be more advantageous, in terms of reconstruction quality and reduced number of radiographic images[8], [9]. These advanced techniques are made possible by GPU-accelerated reconstruction [10]. However, existing imaging systems rigidly fix the source and sensor, using either a curved metal arm or putting both on a circular sliding track. These solutions are bulky and inflexible in terms of medical imaging and desired views/dose

concerns. To expand the possible viewing gamut, it is advantageous to move the source and sensor to separate robotic manipulators, for which there is precedent[11], [12].

### **Manipulator Design**

No manipulator exists with the degrees of freedom and payload capacity necessary to position the x-ray source under a surgical operating table without excessive bulk. Thus, a custom manipulator was designed, using off-the-shelf motors and actuation stages. This under-table robot (UTR) had to be carefully designed, minimizing moment arms and necessary motor torques, while still retaining the flexibility required by the imaging algorithms. The manipulator was characterized for stiffness and overall positional accuracy. The accuracy of the robots is important because it contributes to the 3D reconstruction error, and reconstruction error contributes to pedicle screw placement error.

The Mitsubishi PA10 robot has already been the subject of an accuracy study [11]. This study showed that after correcting robot geometry and stiffness parameters, the mean PA10 position error drops to roughly 0.33mm. A validation protocol for the UTR has been evaluated using aCoordinate Measuring Machine (CMM). The results of the validation protocol are reported in Chapter 2. The results show a RMS position error of 0.248mm and peak error of 0.507mm for the fully loaded UTR with x-ray tube payload. Errors of this magnitude are considered to be of little to no effect on the visualization of fine details post-reconstruction [13].

Manipulator end-effector speeds are low in this application, as the target does not move appreciably. Maximum accelerations are estimated at  $2.5\text{m/s}^2$ , with maximum end-effector velocities of 0.25m/sec. Thus, dynamic constraints are less of a concern, and static accuracy is paramount.

This new under-table robot must be moved in coordination with a Mitsubishi PA10 manipulator, with end effectors locked together by a virtual rigid link. Control of the Mitsubishi

PA10 has already been performed and validated [11]. Thus, Chapter 3 will focus on the control of the UTR and PA-10, and coordination of the two robots. A validation protocol has been performed, using a CMM.

### **Trajectory Planning**

Research at Stony Brook University has resulted in a reconstruction algorithm that allows for 3-D image reconstruction without full 180 degree sweeps of the x-ray equipment[8]. Instead, the algorithm uses prior knowledge of the anatomy to choose view that provide the greatest information for the reconstruction. The order in which these views are collected is unimportant to the reconstruction algorithm, but can make a significant impact on imaging time. For example, if one needed to visit New York, Chicago, and San Francisco, it would not make sense to visit Chicago first or last, as it lies in the middle of the other two cities. This problem is trivial for the example with 3 locations, but the reconstruction algorithm requires 30 or more views. This is the classic traveling salesman problem, and a framework to calculate the cost matrix based on the manipulator dynamics and sort the views such that imaging time is minimized has been completed and tested, as reported in Chapter 4.

### **Reconstruction Sensitivity**

In Chapter 5, the sensitivity of CBCT reconstruction algorithms to source/sensor positioning errors is evaluated common imaging metrics including relative error, correlation coefficient, modulation transfer function, and intensity plots. These metrics will be evaluated for nominal reconstructions, and several reconstructions whose forward projection positions and orientations have been modified by random error of varying magnitudes before reconstruction. Some example cases are provided for the reconstruction of a head phantom. Finally, the sensitivity of this reconstruction algorithm will be compared against the results of the UTR

positioning accuracy study performed in Chapter 2 to evaluate the feasibility of the manipulator/algorithm combination.

## CHAPTER 2 MANIPULATOR DESIGN

As described previously, no robotic manipulator exists that meets the size and performance goals for the x-ray source portion of this imaging system. Thus, a 5-DOF custom robotic manipulator was designed to minimize motor torque demands, while meeting stiffness and accuracy requirements. This manipulator has to fit under a standard Jackson-style operating table, while carrying a 25kg payload and being able to extend above the operating table for lateral view imaging.

### **The Goal**

*The goal of this section is to design, build, and evaluate a new manipulator with the purpose of accurately positioning an x-ray source under static and dynamic conditions.*

### **Introduction**

The importance of geometric calibration of radiographic systems has been well demonstrated [14]. Jeffray et al. demonstrated that geometric calibration of errors as small as 2mm peak magnitude can result in improvements to the reconstruction. Many researchers have developed nonlinear geometric calibration models for robotic systems. In fact, the PA-10 robot used in this application was thoroughly evaluated by a prior U.F. PhD student [11]. This model was significantly more complicated due to the presence of harmonic drives, higher motor torques, errors propagating through serial links, etc. A UTR based upon stepper motors and drives will not have non-linearity due to gearbox backlash, but will lack high-accuracy knowledge of link geometries and joint flexibilities [15]. Link geometries are CAD-modeled and machined to exacting tolerances, but assembly tolerances will yield robot geometric parameters that deviate from nominal values. Additionally, links can be modeled using FEA (finite element analysis) to provide expected link stiffnesses, but bearing and joint flexibilities are only specified

to be less than a maximum tolerance. For these reasons, it is equally important to characterize actual geometric uncertainties and joint flexibilities. Research performed on a prismatic/revolute (PR) gantry assembly [16] deals with quantifying repeatable error due primarily to deflection and manufacturing tolerances in a 6-DOF positioner similar to the 5-DOF positioner being considered. The 6-DOF apparatus uses a prismatic/prismatic/prismatic/revolute/revolute/revolute (PPPRRR) configuration, while the UTR uses a prismatic/prismatic/revolute/prismatic/revolute (PPRPR) configuration. However, the approach provides useful guidance because it focuses specifically on errors resulting from linear positioners.

Additional consideration of error sources showed positional error due to thermal expansion is not negligible. Consider a robotic manipulator with 1 meter reach constructed entirely of aluminum. If the room temperature increases from 20°C to 25°C, a change in length of 200 microns will be observed. This is not insignificant, as overall error is desired to be 1mm or less for both manipulators[13]. The body of research dealing with thermal expansion modeling in robots is limited, but authors have addressed the issue [17]. Thus, temperature sensors are necessary for best accuracy in application, but ignored here as all measurements will be conducted under climate controlled conditions.

## **Methods**

A 5-DOF custom robotic manipulator was designed to minimize motor torque demands, while meeting stiffness and accuracy requirements. The basic design for the manipulator was built on a standard X-Y stage, composed of two prismatic joints. A rotary stage was placed on top of that table, and a vertical, or Z-translation prismatic stage on top of that. Finally, the tube mount was made to be able to rotate about the tube's long axis. The entire manipulator can be seen in Figure 2-1. Only 5 degrees of freedom are necessary to point the x-ray source, as the

detector robot has the ability to rotate the image plane in register with the x-ray aperture/collimator. Other goals include fitting under a standard Jackson-style operating table, while carrying a 25kg payload and being able to extend above the operating table for lateral view imaging. In order to keep eccentric loads on the rotary stage within limits, a counterbalance is necessary, but not shown in Figure 2-1. This brings the total load increase on joints 1-3 to 56kg.

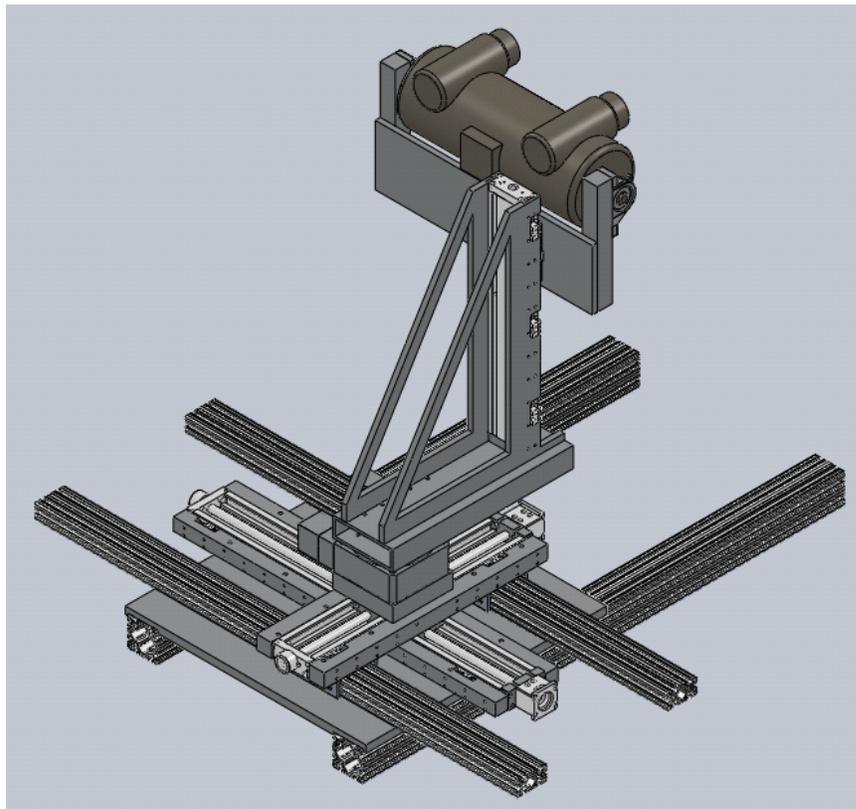


Figure 2-1. SolidWorks model of UTR

Since it was identified that structural stiffness/encoder accuracy is paramount, extra attention was focused on these aspects. The linear actuators chosen have accuracy ratings of  $154\mu\text{m}$  (x-axis) and  $114\mu\text{m}$  (y and z axes), and quadrature encoder resolution of 5 microns. The target is less than 1mm of positional error, to be verified by a CMM (Coordinate Measuring

Machine). Static identification is most important, as the manipulator will be operating in quasi-static conditions, with relatively low accelerations and velocities.

Stepper motors were chosen due to low cost, and suitable torque curves. Servo drive motors produce significantly less low-speed torque for their size, instead spinning at high RPMs to produce similar power outputs. Thus, for this application, servo drives would require expensive low-backlash gear reduction drives, and complicated control algorithms to counteract the nonlinearity of backlash [18]. Additionally, stepper drive motors can be run open loop, and are tolerant of encountering unexpected obstacles. High rotational resistance results in missed steps, which are flagged in the controller by their signature back-EMF sensed by the drive electronics. This does not damage the motor, while servomotors are prone to damage from output shaft lockup. Even though these stepper motors can be run in open loop with guaranteed accuracies due to their inherent design, the apparatus was designed with encoders at each joint to ensure joint position during experimental validation.

One drawback to stepper motors is that they must be sized appropriately and controlled with an acceleration profile developed from their speed/torque curves. That is, the acceleration limit for the motor is a function of the motor's current velocity. Figure 2-2 shows this phenomenon, for the smaller of the two types of motors used. The motors were powered with 48VDC, for which torque curves were not provided by the manufacturer. They can be estimated from the 24VDC and 36VDC curves provided.

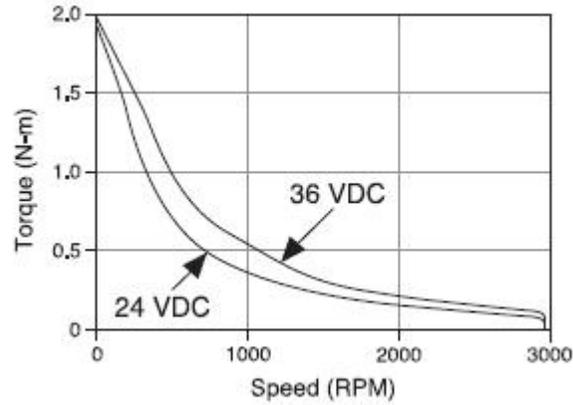


Figure 2-2. Available torque as a function of velocity [19]

These calculations have been performed for all 5 axes, with an assumed payload of 1.5 times that which is expected in the design. Motors were selected to give similar end-effector velocities to that of the PA-10 manipulator, though clinical imaging will occur at speeds less than 1/10<sup>th</sup> maximum end effector velocity.

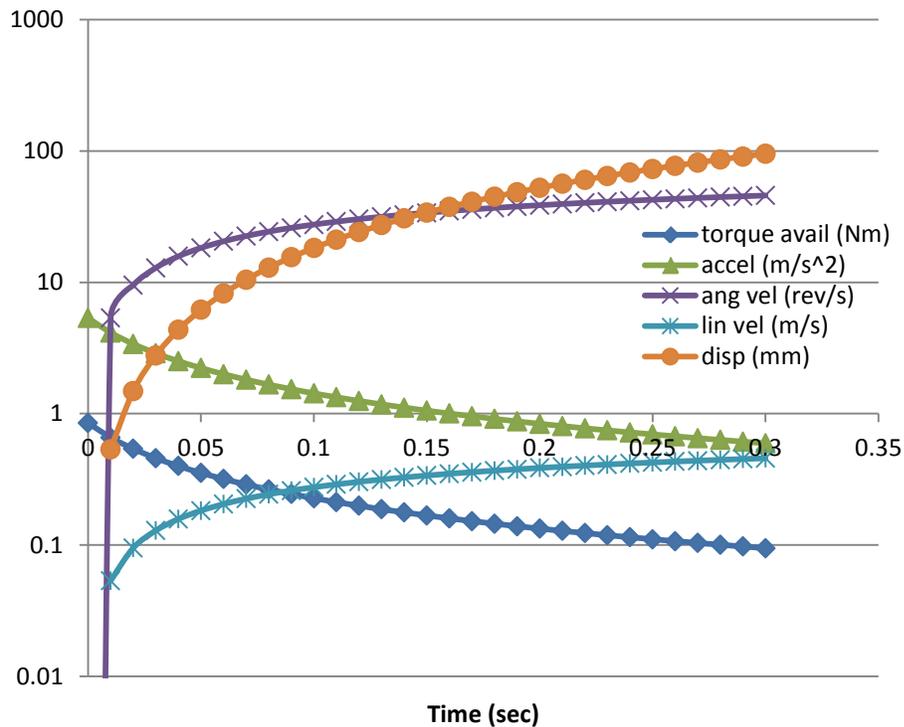


Figure 2-3. Joint 2's position, velocity, acceleration, and torque as a function of time

A sample result of these dynamics calculations is shown in Figure 2-3. This calculation was performed to find the position, velocity, and acceleration as a function of time, if maximum acceleration is commanded.

Although significant work was applied in the initial design of the UTR, there is still link flexibility and manufacturing/assembly tolerances that can be modeled and quantified to improve end-effector accuracy.

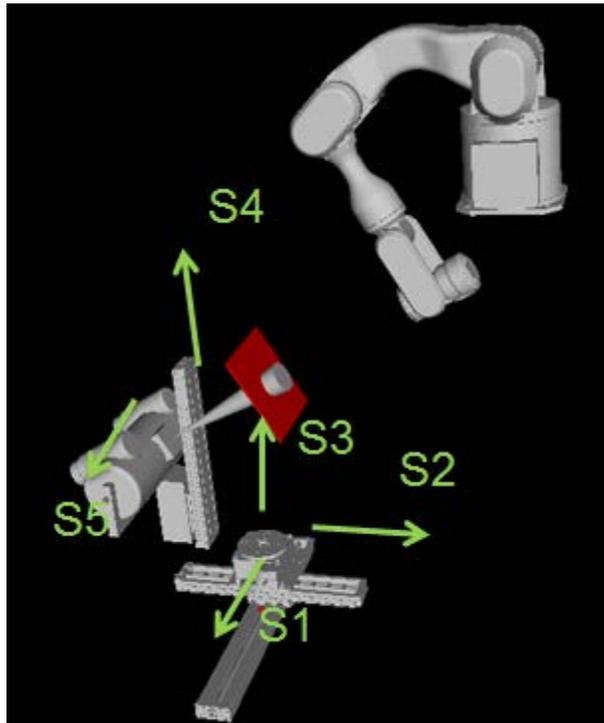


Figure 2-4. Labeled joint axes of the UTR

The forward kinematics for the UTR are straightforward.

$${}^{EE}_F T = \begin{bmatrix} -s_3 s_5 + c_3 c_5 & -c_5 s_3 - c_3 s_5 & 0 & s_1 + .15c_3 \\ c_3 s_5 + s_3 c_5 & -s_3 s_5 + c_3 c_5 & 0 & s_2 + .15s_3 \\ 0 & 0 & 1 & s_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Where  $S_1, S_2$ , and  $S_4$  are the z, y, and z prismatic joint positions as shown in Figure 2-4, and  $\theta_3$  and  $\theta_5$  are the revolute joint angles about their respective joint axes, also shown in Figure 2-4. Finally,  $s_i$  and  $c_i$  are shorthand for the sin and cosine of joint  $i$ .

Similarly, we find the inverse kinematics are also straightforward.

$$\mathbf{R} = {}^F_P \mathbf{T} \begin{bmatrix} 0 \\ 0 \\ SID/2 \\ 1 \\ R_3 \end{bmatrix} \quad (2.2)$$

$$\theta_5 = \tan^{-1} \frac{R_3}{\sqrt{R_2^2 + R_1^2}} \quad (2.3)$$

$$\theta_3 = \tan^{-1} \frac{R_2}{R_1} \quad (2.4)$$

$$S_1 = R_1 + T_{1,4} - L_{arm}c_3 + L_{tube}S_3 \quad (2.5)$$

$$S_2 = R_2 + T_{2,4} - L_{arm}s_3 - L_{tube}c_3 \quad (2.6)$$

$$S_4 = R_3 + T_{3,4} \quad (2.7)$$

The transformation of the desired image plane,  ${}^F_P \mathbf{T}$ , is known. Using that transformation matrix, then a vector is projected that is the desired central ray. From there, it is a matter of evaluating the tube elevation and azimuth, then finding the appropriate positions for the prismatic joints.

## Implementation and Experimental Assessment

In order to measure actual end-effector positioning, a FaroArm Silver (Faro Technologies, Lake Mary, FL), is rigidly mounted to the robot frame. A FaroArm is a multi-link measurement device with an arm and tooling ball, or probe which is freely positioned by hand. Typically, the tooling ball is run along a groove or other marker on the object, creating a point cloud that can be later analyzed. Although the FaroArm uncertainty is higher than that of the

other methods (127 micron), it offers the ability to check multiple links at once, narrowing down the source of uncertainty and providing useful feedback for design improvements.

The transformation from the robot base to the robot end-effector is defined as follows:

$${}_{EE,UTR}^{BASE}T = {}_2^1T \cdot {}_3^2T \cdot {}_4^3T \cdot {}_5^4T \cdot {}_{EE}^5T \quad (2.8)$$

Each transformation matrix represents the kinematic change from joints 1 through 5. The product of these 5 transformations provides the forward kinematics for the UTR [20]. This transformation can be approximated by the design parameters for each link, but cannot be known exactly. The deviations from design parameters are the desired outputs from the calibration process [21].

The FaroArm is defined similarly, with 6 revolute links. The joint space to task space conversion is handled by the FaroArm itself – only a point in XYZ coordinates is reported. This point is given in the FaroArm coordinate system, which is not aligned with the coordinate system of the UTR. Thus, another unknown transformation exists:

$${}_{BASE,UTR}^{FAROARM}T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & Tx \\ R_{21} & R_{22} & R_{23} & Ty \\ R_{31} & R_{32} & R_{33} & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

This transformation matrix is initialized by moving the robot to 7 positions along its x-axis, and a least-squares line fit is defined as,  ${}_{BASE}^{CMM} \bar{x}$ . Similarly, the robot is moved to 7 positions along its y-axis, and a least-squares line fit defines  ${}_{TEMP}^{CMM} \bar{y}$ . Then, the cross product of these two vectors is defined as the robot z-axis in the CMM system,  ${}_{BASE}^{CMM} \bar{z}$ . In the last step, the y-axis of the robot, orthogonal to the basis vectors  ${}_{BASE}^{CMM} \bar{z}$  and  ${}_{BASE}^{CMM} \bar{x}$  is defined as  ${}_{BASE}^{CMM} \bar{y}$ . The rotation matrix from FAROARM to BASE,UTR is initialized from these three vectors. The translation vector from FAROARM to BASE,UTR is constructed by direct measurement of the robot while in the home position (all joint values equal to zero). Thus, the transformation  ${}_{BASE,UTR}^{FAROARM}T$  is initialized.

Next, a set of 50 joint-space positions is generated for the robot. These 50 views cover the robot’s workspace that would be used during the collection of x-ray images. Using the MATLAB uniform random number generator, each joint space is sampled evenly. The resulting task-space positions are shown in Figure 2-5.

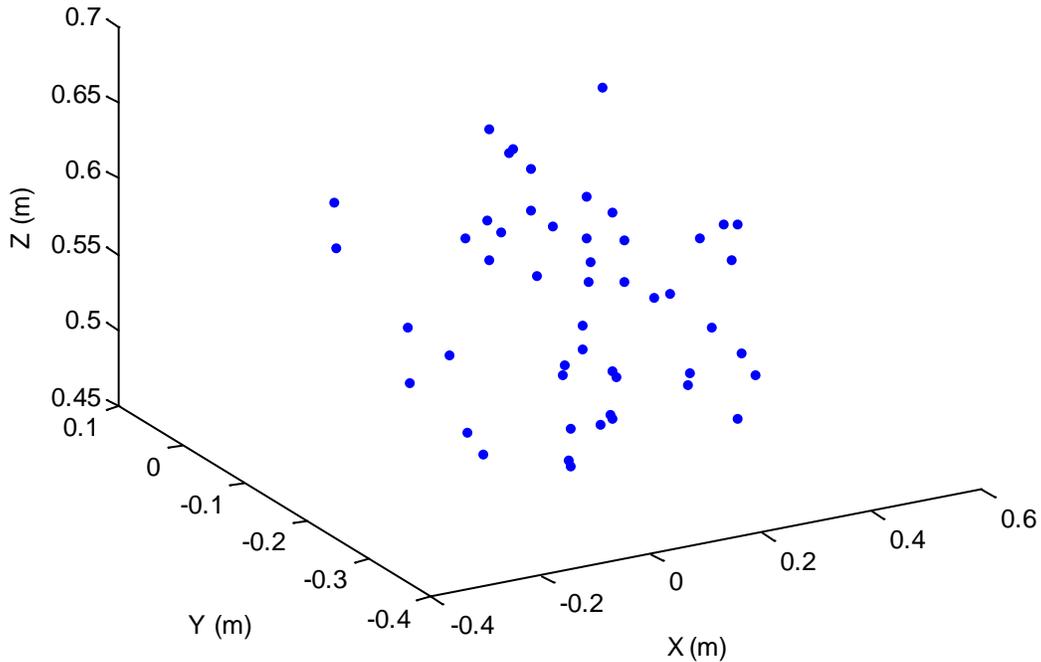


Figure 2-5. A set of 50 unique positions

A trust-region-reflective nonlinear least squares optimization was implemented in MATLAB (Mathworks, Natick, MA) to minimize the cost function:

$$E = \sum_{i=1}^{40} (\vec{x}_{i,d} - \vec{x}_{i,m})^2 + \sum_{n=1}^{14} (\Delta p_n)^2$$

In the function above,  $\vec{x}$  is defined as the x, y, and z components of the i-th view. The subscript m refers to the robot fiducial position as measured by the FaroArm. The subscript d refers to the desired position, of the robot end-effector. Both are measured in the FaroArm

coordinate system. Finally,  $\Delta p_n$  is the variation in each DH parameter. They are included in the cost function in order to drive the variation out of robot parameters and into the coordinate transform between the FaroArm and the UTR, where applicable.

## **Experimental Setup**

The measurement device in the experimental setup consists of a FaroArm model IND-03. It is fixed rigidly to the frame of the UTR robot, and is equipped with a ball probe for measurement of 1/8” holes in the robot end-effector. The FaroArm was not designed to be used with LabVIEW, so its proprietary serial communication protocol was reverse engineered. Data points (X,Y,Z) can be collected at up to 50 per second. This is a limitation of the 57600 baud serial port.

The UTR robot consists of two linear stages in the X and Y directions, a rotary stage about the Z axis, and a linear stage along the Z axis. The fifth joint is part of the payload, and it controls tube elevation. The robot is equipped with stepper motors configured to 2000 steps/rev, resulting in a resolution of 0.005mm/step on the linear stages, and 0.0025deg/step on the rotary stage. The fifth axis uses a direct-drive stepper motor, with resolution 0.18deg/step. This can be improved further with the addition of a low-backlash gear reduction drive, but the orientation of the tube does not impact the reconstruction quality as long as the focal point of the x-ray tube is on the axis of rotation. These steppers are run in open loop, though the system is equipped with encoders. Stepper motors are typically not for use in closed-loop control. The stepper motors are outfitted with back-EMF detection to identify missed steps due to unexpected dynamics or obstacles [22]. The robot is controlled using National Instruments’ LabVIEW, coupled with five NI 9512 1-axis motion controllers and five P70530 stepper motor drives. The motors themselves are coupled to the joints with low backlash spline-type couplers. The stages are equipped with home and limit switches, the former defines the zero-point for each joint.

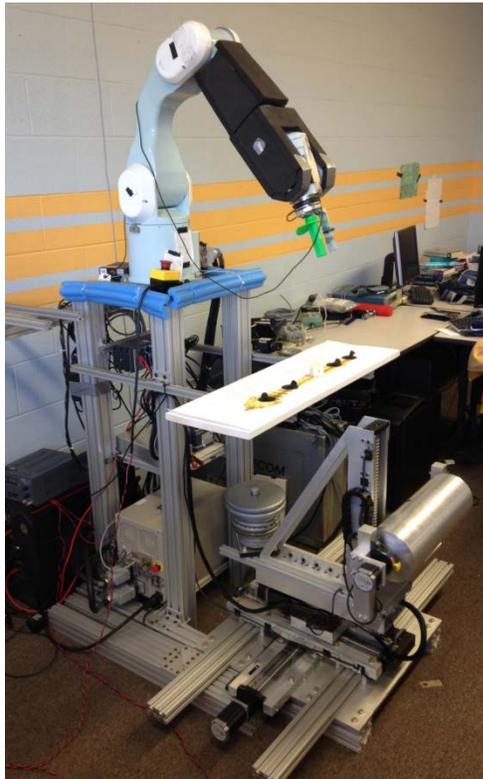


Figure 2-6. Experimental setup showing PA10 and UTR

The experimental setup, as shown in Figure 2-6, was used to collect seven datasets to quantify UTR positioning error. These datasets are described in Table 2-1, and will be referred to in the results. The positions of the first four datasets were visited in order of increasing specific joint position (X,Y, Z-rot and Z-trans). The next three datasets were visited in order of increasing joint 3 position. The volume covered in the 50 position dataset is shown in Figure 2-5.

Table 2-1. Datasets collected with the experimental system

Name	Number of Positions
Robot_Xaxis	7
Robot_Yaxis	7
Robot_Zrot	13
Robot_Ztrans	15
Robot unloaded	50
Robot 5kg load	50
Robot 28kg load	50

Before analyzing data, it is important to examine the instrument being used to measure the data, and check for any impact on results. Ideally, the measurement device creates zero impact on the test system, or the impact is at least measurable. For example, a typical voltmeter has a resistance in the mega-Ohm range, large enough to have little effect upon most circuits. The FaroArm works by placing the ball probe in a feature, guiding the probe by hand. The feature, a hemisphere equal in size to the ball probe, was positioned such that the probe force to engage the hemisphere is downward, with gravity. This is because the manipulator is stiffest in this direction, resulting in the least impact on measurements as a result of the force imparted by the FaroArm probe.

Using a randomly selected 40 of the 50 positions measured, the nonlinear least squares optimization described earlier was performed. The remaining 10 positions are used to validate the calibration. A total of 22 variables are used to represent the deviations from nominal values. Six of these deviations are used in finding  ${}^{FAROARM}_{BASE,UTR}T$ , the transformation between the coordinate systems of the FaroArm and UTR.

The RMS and peak errors were calculated by the following formulas:

$$Error(i) = \sqrt{(x_{i,d} - x_{i,m})^2 + (y_{i,d} - y_{i,m})^2 + (z_{i,d} - z_{i,m})^2} \quad (2.10)$$

$$RMSError = \sqrt{\frac{1}{50} \sum_{i=1}^{50} Error(i)^2} \quad (2.11)$$

$$Peakerror = \max(Error) \quad (2.12)$$

Where  $i$  denotes the position number,  $x_{i,d}$  denotes the current desired position, and  $x_{i,m}$  denotes the current measured position. All 50 positions were used in the error metric calculations.

## Results

Figure 2-7 shows the results of one of many FEA studies performed on the UTR. In this case, the end effector is loaded with a force of 125N, which is representative of a  $2.5\text{m/sec}^2$  acceleration. Additionally, it is loaded with force due to gravity. This acceleration is the maximum available with the selected motors. The maximum deflection at the end effector in this loading configuration is 140 microns.

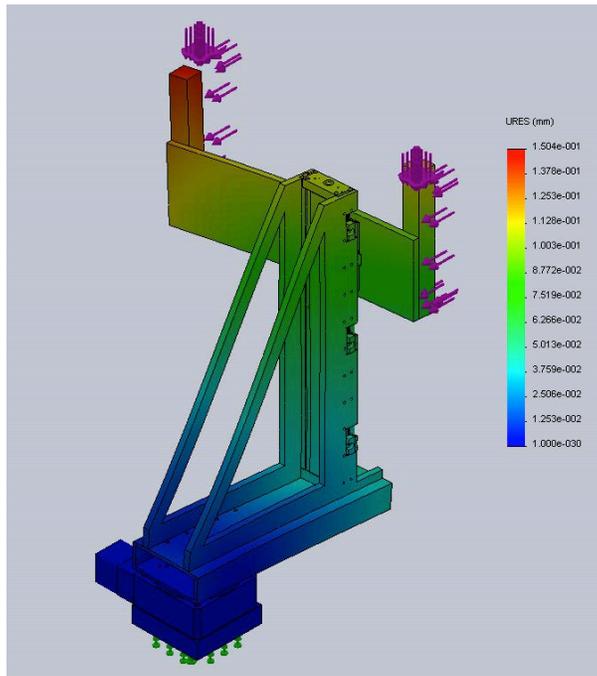


Figure 2-7. Solidworks FEA model of typical UTR arm loading during acceleration

The worst case FEA model shows deflection of 250 microns, at maximum acceleration. Typical accelerations will be roughly 20% of maximum accelerations, to maintain reasonable joint velocities for the operating room environment.

The first 6 deviations are used to calibrate the transformation matrix between the FaroArm and the UTR base (Table 2-2). The remaining 16 deviations are used to calibrate the first 4 joints of the UTR (Figures 2-8 through 2-10). The deviations used to calibrate the

geometric model of the UTR had deviations with magnitude less than 1 deg or 1mm (Tables 2-3 through 2-6). Joints 1 through 5 are depicted in Figure 2-4.

Table 2-2. Changes to  ${}_{BASE,UTR}^{FAROARM}T$  as a result of calibration

	Rotation (deg)	Translation (mm)
X-Axis	-0.0605	-2.5296
Y-Axis	0.1290	0.4645
Z-Axis	0.4975	-0.4234

Table 2-3. Original D-H parameters for the UTR

	theta (deg) joint angle	d (mm) link offset	a (mm) link length	alpha (deg) link twist	offset from home switch
Joint 1	-pi/2	0	0	-pi/2	0
Joint 2	-pi/2	0	0	pi/2	0
Joint 3	0	0	200.0000	0	0
Joint 4	0	0	0	pi/2	500mm
Joint 5	0	-75.0000	0	-pi/2	0

Table 2-4. Modifications to the D-H parameters for the UTR after calibration, unloaded

	theta (deg) joint angle	d (mm) link offset	a (mm) link length	alpha (deg) link twist	offset from home switch
Joint 1	0.0000	n/a	0.0000	-0.1248	0.0000 mm
Joint 2	-0.1051	n/a	0.0000	-0.0261	0.0000 mm
Joint 3	n/a	0.0000	0.0735	-0.0054	0.0000 deg
Joint 4	-0.0001	n/a	0.0000	-0.0000	0.0000 mm
Joint 5	n/a	n/a	n/a	n/a	n/a

Table 2-5. Modifications to the D-H parameters for the UTR after calibration, 5kg load

	theta (deg) joint angle	d (mm) link offset	a (mm) link length	alpha (deg) link twist	offset from home switch
Joint 1	0.0000	n/a	0.0000	-0.1046	0.0000 mm
Joint 2	-0.0907	n/a	0.0000	-0.0430	0.0000 mm
Joint 3	n/a	0.0000	0.0390	-0.0198	0.0000 deg
Joint 4	0.0001	n/a	0.0000	0.0000	0.0000 mm
Joint 5	n/a	n/a	n/a	n/a	n/a

Table 2-6. Modifications to the D-H parameters for the UTR after calibration, 28kg load

	theta (deg) joint angle	d (mm) link offset	a (mm) link length	alpha (deg) link twist	offset from home switch
Joint 1	0.0000	n/a	0.0000	-0.0829	0.0000 mm
Joint 2	-0.0951	n/a	0.0000	-0.1234	0.0000 mm
Joint 3	n/a	0.0000	-0.2203	-0.0468	0.0002 deg
Joint 4	-0.0006	n/a	0.0000	0.0000	0.0000 mm
Joint 5	n/a	n/a	n/a	n/a	n/a

Note that the D-H parameters listed in Table 2-4 are for the unloaded case. Because the payload of the UTR is the fifth axis, the Joint 3 link length and Joint 5 link offset will change when the payload is installed. Joint 3 gets longer (a longer effective arm) and the offset along  $S_5$  changes as well. Therefore, it is not meaningful to compare Joint 5's link length change in the 28kg configuration with the unloaded and 5kg load configuration link length changes. Because the offset along  $S_5$  is just a property of the chosen probe point, it is not listed. With an actual x-ray tube, this distance would be the offset to the focal point of the tube.

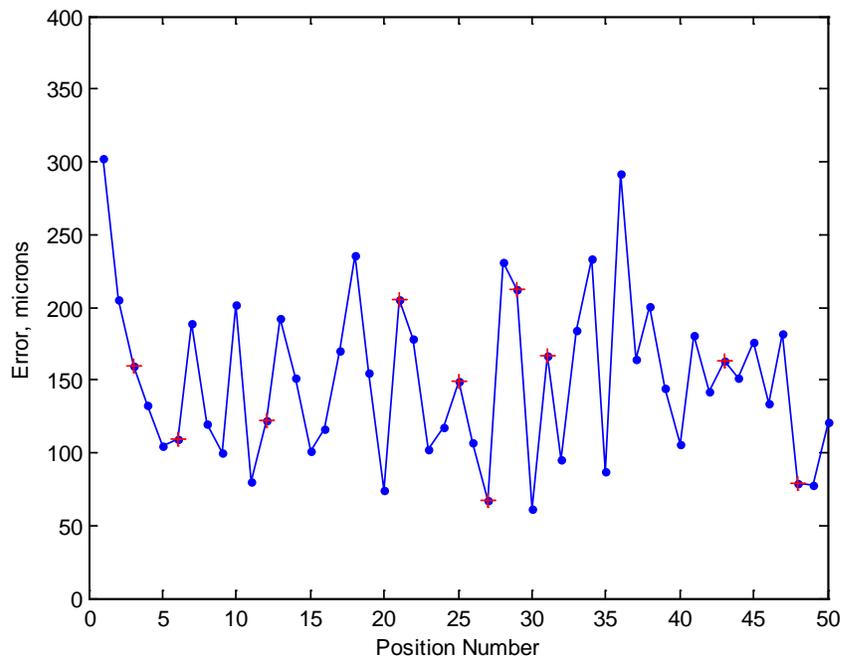


Figure 2-8. Position error for unloaded case. Red points are not included in the optimization, and are present to verify the calibration. All points are included in reported error metrics.

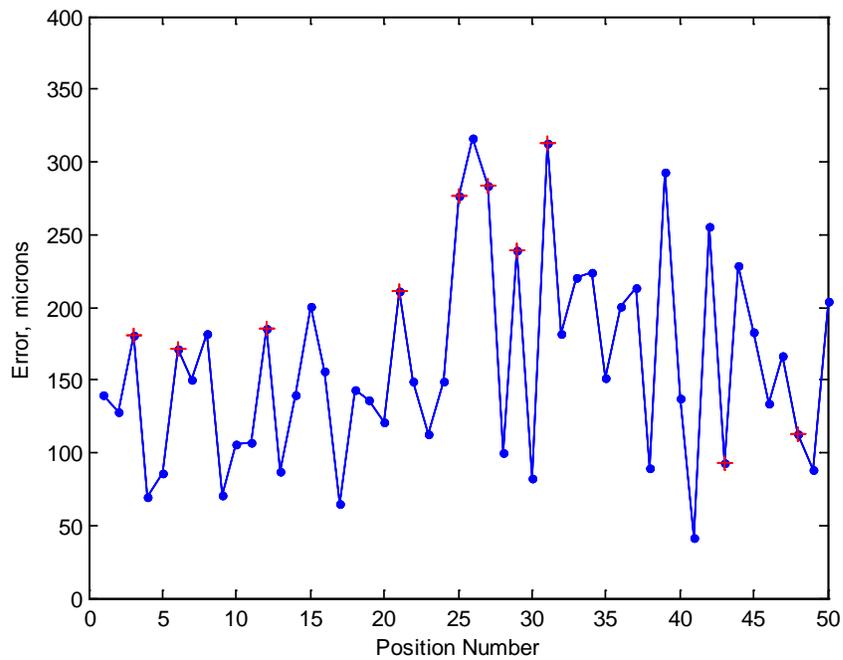


Figure 2-9. Position error for 5kg load case. Red points are not included in the optimization, and are present to verify the calibration.

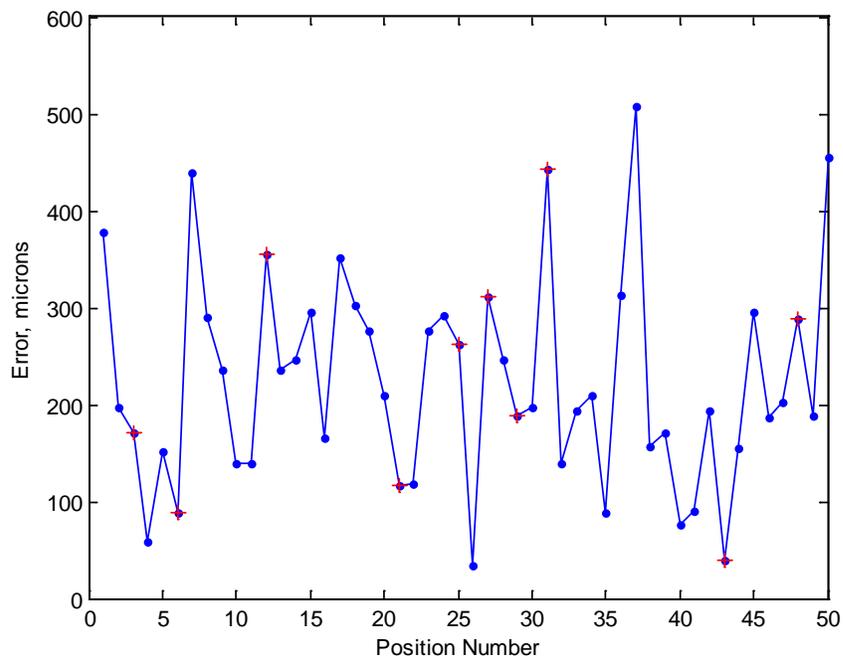


Figure 2-10. Position error for 28kg full payload case. Red points are not included in the optimization, and are present to verify the calibration.

Table 2-7. Error of calibrated model

Dataset	RMS Error (mm)	Peak Error (mm)
UTR 50 point unloaded	0.160	0.303
UTR 50 point 5kg load	0.174	0.316
UTR 50 point 28kg load	0.248	0.507
PA-10 unloaded [11]	0.32 (mean)	0.66

Table 2-7 shows the RMS and peak errors for the calibrated geometric models produced by each dataset. The 28kg load reflects the load on the end-effector. An equal counterbalance is necessary, bringing the total load increase on joints 1-3 to 56kg.

### Discussion

The UTR was demonstrated to have the necessary positioning capabilities to accurately position the x-ray source. Specifically, the UTR with full payload was shown to have RMS total positioning errors of 0.248mm RMS and 0.507mm peak over the sampled workspace.

Furthermore, the UTR was shown to have the spatial capabilities necessary to perform the imaging tasks while not interfering with fixed operating room equipment. The results of Chapter 5 demonstrate that this level of positioning error has relatively insignificant effects on image reconstruction.

Relative to current Iso-C arm and O-arm systems, the proposed system also provides enhanced usability. By moving the imaging equipment to separate robotic manipulators, equipment can be moved out of the way to gain access for a particular procedure. Additionally, it is possible to capture almost any single view, if the surgeon desires confirmation of a particular screw placement, for example. Both of these tasks can be controlled through closed-loop force feedback, resulting in an easy to use platform for surgical imaging and navigation. Additionally, the use of 5 and 6 DOF robotic manipulators permits non-circular imaging trajectory for advanced Cone Beam Computed Tomography (CBCT) reconstructions.

By removing the circular imaging constraint, improvements in patient dose and/or reconstruction quality can be realized by imaging only the most salient views [8], [23], [24]. With a priori knowledge of the area to be imaged, the most informative views can be identified. Frequently, these views lie outside of a circular trajectory, and therefore must be disregarded due to current limitations in imaging hardware. Moving the imaging hardware to two separate robots increases the available workspace significantly, permitting the desired radiographic views to be captured.

With positioning errors less than 1mm, it is possible to perform good quality reconstructions using standard Feldkamp-Davis-Kress (FDK) framework. With more advanced algorithms such as iterative and algebraic methods, off-axis views can be utilized to further improve image quality for given radiological dose, or reduce radiological dose for a given image quality [8]. Future steps include a study of the surgeon-robot interaction (Chapter 3), a study of optimal trajectories to gather a set of views (Chapter 4), and finally a study of the impact of positioning error on image reconstruction (Chapter 5).

### **Conclusion**

Through the development of this manipulator, a new surgical imaging concept has been introduced. By moving the x-ray source and sensor to separate robotic manipulators, improvements in surgical ergonomics and workflow, radiological dose and image reconstruction quality can be realized. The early prototype shows travel speeds and positioning accuracies that demonstrate the feasibility of the concept.

## CHAPTER 3 MANIPULATOR CONTROL AND COORDINATION

Since a dynamic radiographic imaging system requires precise alignment between the source and sensor for quality image reconstruction, accurate control of the imaging robots is important. The dynamics and control laws for the Mitsubishi PA-10-6CE have been well explored by researchers here at the University of Florida [11], [12], [25] and elsewhere. This work will be relied upon, but not recreated in this project. Instead, this project will focus on new, additions to existing tools within the Dynamic Robotic Imaging Control Software (DRICS) platform [26], and the integration of the UTR.

### **The Goal**

*The goal of this section is to develop and evaluate the controllers for both the pre-planned trajectory mode and the surgeon-interactive (haptic) mode. This includes controllers for both the PA10 and UTR robots.*

### **Introduction**

In order to perform robotic surgeon-interactive manipulator based imaging, two separate controls tasks are necessary: autonomous control and haptic control. Autonomous control is needed for the collection of images, the position of which is predefined by an offline path planning algorithm. Dozens of control schemes have been created and published, but they can be generally classified as linear or nonlinear in approaches. The well-known PID (proportional-integral-derivative) controller [27] is probably the most popular of the linear control schemes, and the RISE (robust integral of the sign of the error) control scheme [28] is a recently developed nonlinear control scheme with promising results for robotic manipulator control.

Additionally, there exists a need for haptic control – that is, control of the robotic manipulator by direct interaction. This serves two purposes: guiding the manipulator into

position, and feedback to the surgeon of joint limits and other boundaries. Haptic controllers can be generally classified as impedance controlled devices or admittance controlled devices, or direct force feedback devices (explicit force control) [29][30]. A device under impedance control behaves as follows: the operator moves the device, and the device reacts with a force if a virtual object is met. Therefore, the operator feels the mass and friction of the device, so the manipulator must be low inertia and low friction (i.e. backdrivable) in order for effective control to take place. This control scheme is suitable for lightweight, cable driven manipulators such as the RIO (MAKO Surgical, Davie, Florida). On the other hand, admittance control is the inverse of impedance control. A device under admittance control behaves as follows: the operator applies a force to the haptic device, and the device reacts with the proper displacement. Advantages of admittance control include considerable freedom in the mechanical design of the device, e.g. having high inertia and friction. As a result, the device can be very robust with high stiffness and capable of exerting high forces and torques.

In order to be a good candidate for impedance control, the robot must have relatively low joint friction. The Mitsubishi PA10 used in this study has Coulumb friction torques equal to 9.08, 9.28, 5.81, 1.48, 1.84, and 0.87 Nm for joints 1-6, respectively [11]. With friction values that high, applied end-effector forces will need to be in excess of 15N for most of the workspace desired. Therefore, impedance control (especially without force feedback) is not suitable, nor is the position control aspect needed for the surgeon-interactive imaging application. Direct admittance control is not suitable either, as there are typically multiple robot configurations to achieve any desired end-effector post and singular configurations will require additional control logic. For instance, if the arm is fully extended and then force is applied to move the end effector inward, joint 3 of the arm could move in either direction. Admittance control relies on the

Jacobian inverse, which becomes ill-conditioned at physical singularities. Although there are strategies for dealing with ill-conditioned Jacobian matrices such as the damped least squares method and null space method [31], admittance control is not a good fit for this application.

Therefore, gravity-compensated explicit force control will be evaluated. In explicit force control, forces and torques are applied to the load cell mounted on the end-effector, and these are then resolved through the 6 manipulator joints. Torques are applied at each of these six joints with direction and magnitude proportional to the applied forces and torques, in order to attempt to minimize the applied force [32].

### Methods

In conventional (position-based) control of a serially linked servo-powered robotic manipulator, the block diagram looks like this:

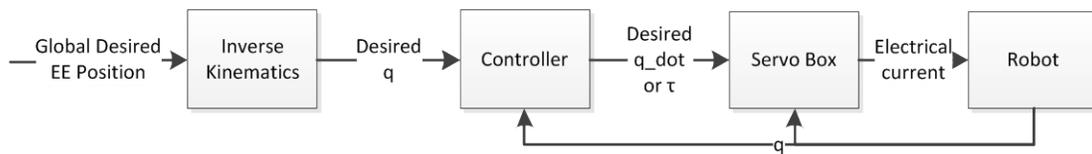


Figure 3-1. Block diagram of PA10 control.

The controller provides desired torques or velocities to the robot’s servo box. The PA10 accepts either command type, though torque commands bypass one of the internal servo box control loops, and allow for a greater degree of control (we cannot modify the velocity control loop parameters within the servo box).

### Pre-planned Control

Although haptic control is necessary for surgeons to be able to define specific views and to move the robots out of the way, pre-planned autonomous trajectory control is necessary to gather the fluoroscopic views that form the basis for 3-D reconstruction. The trajectory is

discretized at a finite step, and both robots are given position targets along a path. These targets are continually modified, forming a trajectory.

In autonomous control mode, we use Crane's analytic method for inverse kinematics of the PA10 to convert from task space to joint space [33]. There are numeric methods, but as is the case with most numeric methods, they do not always converge to the best solution and are often more computationally complex. For the PA10, there are up to 8 possible configurations/solutions. Numeric approaches typically only return one of these - the closest one to the initial guess set, in the direction that the optimizer travels first. Therefore, the analytic approach is better.

Inverse kinematics solvers fail when a "singularity" is approached. This happens when two joints line up, which can happen with joints 1, 4, and 6 on the PA10. In this case, the number of unique solutions goes from 8 to infinity. Although singularities typically will not be encountered in the PA10 workspace for imaging tasks, the condition number of the manipulator Jacobian can be used to monitor "closeness" to singular configurations.

$$c(J) = \|J\| \|J^{-1}\| \quad (3.1)$$

As the condition number increases, the manipulator approaches a singularity. We can simply "drive through" the singularity by holding the previous command until the condition number improves.

Although the PA10 robot uses servo motors equipped with harmonic drives and the UTR uses stepper motors, both can be controlled with velocity commands. The PA10 servo box converts these joint velocities to motor current, using an internal closed loop PI controller. The performance of this controller is excellent at the low velocities encountered in the surgical imaging application. The UTR uses stepper motors, which are driven by a variable frequency

square wave. The stepper-motor velocity is linearly proportional to the square wave frequency as long as the available motor pull-in torque is greater than the motor load. Thus, velocity control is possible in open loop. Encoders are used for real-time confirmation of desired position and velocity, and trip a fault if velocity is not within tolerance (if the apparatus experiences a failure or unexpected object in its path). Additionally, a stall detection circuit utilizing back-EMF measurement is in place, similar to the method described by Bui et al. [22]. Thus, open loop control of the UTR is possible, as long as commanded velocities and accelerations are within the dynamic limits, governed by the available motor torque [34]. Because stepper motors are capable of open loop position and velocity control, the control task is relatively simple.

Conversely, the servo motors of the PA10 require closed loop control. Two methods of control are evaluated: linear (dual-loop digital P/PI, [27]) and nonlinear (RISE [28]). The dual-loop P/PI controller consists of an outer position loop running on a realtime target, the CompactRio 9024 (National Instruments, Austin, TX). Typical loop times are 1000 $\mu$ s for the outer loop. The inner velocity loop runs on the servo drive provided by the manipulator manufacturer. Loop time for the velocity loop is 665 $\mu$ s. Finally, there is an analog control loop for joint current, also on the manufacturer's servo drive. This analog control loop can accept torque commands directly, bypassing the aforementioned digitally implemented control loops.

The RISE control law calculates torque commands with the following equation:

$$\tau = (k_s + 1)e_2 + \int_0^t [(k_s + 1) \alpha_2 e_2 + \beta_1 \text{sgn}(e_2)] d\sigma \quad (3.2)$$

$$e_1 = q_d - q \quad (3.3)$$

$$e_2 = \dot{e}_1 - \alpha_1 \dot{e}_1 \quad (3.4)$$

where  $k_s, \beta_1, \alpha_1, \alpha_2$  are positive constant control gains. Both of these controllers will be tested for their tracking performance in a typical move between imaging views. An instantaneous move of 150mm in task space is executed, and the joint positions and torques are recorded.

### Haptic Control

During manual control of the system, the PA10 assumes the master role, and the UTR attempts to follow. Operation is governed by the following explicit force control law:

$$\boldsymbol{\tau} = f(F_{surgeon}^{EE}, \mathbf{q}, \mathbf{x}_{UTR}) \quad (3.5)$$

Commanded joint torques are a function of the force input on the handle, and optionally the current position of the undertable robot and any possible collisions with the environment. This function is nonlinear, and includes a deadband to eliminate end-effector drift and a non-linear hyperbolic tangent function with tunable parameters to handle joint limits and collision avoidance.

Consider the following, based on virtual work: [27]

$$\mathbf{F}' \Delta \mathbf{x} = \boldsymbol{\tau}' \Delta \mathbf{q} + \mathbf{G}' \Delta \mathbf{q} \quad (3.6)$$

Where:

$\mathbf{F}$  is a vector of Cartesian forces/torques (measured by the load cell at the end effector)

$\Delta \mathbf{x}$  is an infinitesimal (vector) distance in Cartesian coordinates (delta X)

$\boldsymbol{\tau}$  is the set of 6 joint torques

$\Delta \mathbf{q}$  is an infinitesimal rotation (vector) of the 6 robot joints (delta q)

$\mathbf{G}$  is the vector of joint torques due to gravity

The Jacobian is a matrix of partial derivatives between two vectors:

$$\dot{\mathbf{x}} = \mathbf{J} * \dot{\mathbf{q}} \quad (3.1)$$

$\dot{\mathbf{x}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\alpha} \ \dot{\beta} \ \dot{\gamma}]'$  (a column vector of the linear velocity and angular velocity of the end effector) and

$\dot{\mathbf{q}} = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3 \ \dot{q}_4 \ \dot{q}_5 \ \dot{q}_6]'$  (a column vector of joint velocities)

The Jacobian is a function of the geometry of the robot and can be used for force (stiffness) control. The current framework calculates the Jacobian using the method outlined by Paul and Shimano [35]. We see that the Jacobian relates Cartesian space to joint space. Therefore:

$$\Delta \mathbf{x} = \mathbf{J} \Delta \mathbf{q} \quad (3.2)$$

Knowing that, we can manipulate as follows:

$$\mathbf{F}' \mathbf{J} \Delta \mathbf{q} = \boldsymbol{\tau}' \Delta \mathbf{q} + \mathbf{G}' \Delta \mathbf{q} \quad (3.3)$$

We factor and rearrange:

$$\mathbf{F}' \mathbf{J} \Delta \mathbf{q} = (\boldsymbol{\tau} + \mathbf{G})' \Delta \mathbf{q} \quad (3.4)$$

which holds for all  $\Delta \mathbf{q}$ , so we cancel  $\Delta \mathbf{q}$ .

$$\mathbf{F}' \mathbf{J} = (\boldsymbol{\tau} + \mathbf{G})' \quad (3.5)$$

which can be solved for t:

$$\mathbf{J}' \mathbf{F} - \mathbf{G} = \boldsymbol{\tau} \quad (3.6)$$

which relies on the Jacobian matrix transpose, not the inverse of the Jacobian Matrix, so there is no problem even at singularity as we do not need to calculate the inverse [27].

Admittance control relies on the inverse of the Jacobian, while explicit force control relies only on the transpose of the Jacobian. Thus, the problem of a singularity is an issue in admittance control, but not direct force control. We measure  $\mathbf{F}$ , can calculate  $\mathbf{J}$ , can calculate  $\mathbf{Q}$ , so we can find  $\boldsymbol{\tau}$ , which again, represents the desired joint torques. Then, we multiply the control gains by  $\boldsymbol{\tau}$ . This can be thought of this as "increasing the leverage", as this is the degree to which the motors assist. If the gains were zero, the friction at the joints and inertia of the manipulator

would have to be overcome by the force applied at the end effector. If the gains were 100,000, then applying even the slightest force would result in extreme motion. Without any user interaction, there is roughly 1N of noise measured at the load cell. We use gains of 8-15. In this case, and for typical poses, a force of 10N at the load cell results in a torque command of roughly 25N-m, which is about 15% of maximum available torque.

During manual control of the robot, it may be desirable to reduce the control effort based upon a possible collision or joint limit. The control law then becomes:

$$\boldsymbol{\tau} = \mathbf{K}\mathbf{J}^T[\mathbf{S} \circ^{EE} \mathbf{F}_{surgeon}] - \mathbf{G} \quad (3.7)$$

where  $\mathbf{S}$  is a column vector comprised of  $\mathbf{S}_x, \mathbf{S}_z, \mathbf{S}_z$  as described in Equation 3.9,  $\mathbf{K}$  is the vector of control gains, and  $\circ$  is the Hadamard product. Care must be taken to match the coordinate system of  $\mathbf{S}$  with the coordinate system of the robot end effector. This will have the effect of the surgeon feeling increasing resistance at a prescribed limit. The  $F_{surgeon}^{EE}$  is measured by a ATI Mini45 F/T load cell, with 6 degrees of freedom, and a calibrated range of +/- 290 N, +/- 10 N-m. This signal is lowpass filtered by a first-order EWMA (Exponentially Weighted Moving Average) filter, set at 2Hz. This is justifiable since the majority of frequency content of human skeletal muscle lies under 2Hz [36]. The filter is implemented digitally, as follows:

$$y(n) = \alpha \cdot x(n) + (1 - \alpha) \cdot y(n - 1) \quad (3.8)$$

where  $x(n)$  is the current measurement,  $y(n-1)$  is the previous filtered value, and  $y(n)$  is the current filtered value. This filter topology was chosen because it is very simple computationally, and will not tax the real-time processor, it consisting of just two multiplications, an addition, and a subtraction. Using this filter provides gain margin in the control loop by attenuating high frequency signals which otherwise would feed back and resonate at high control gains, resulting in uncontrolled oscillation.

## Haptic Study

In order to test the effectiveness of the haptic control laws, an apparatus was produced composed of a spine model and 8 circular photoresistors (active diameter 2.54mm). One photoresistor is placed at each target shown in Figure 3-2, and one sensor is mounted on each side of the cube shown at T7.

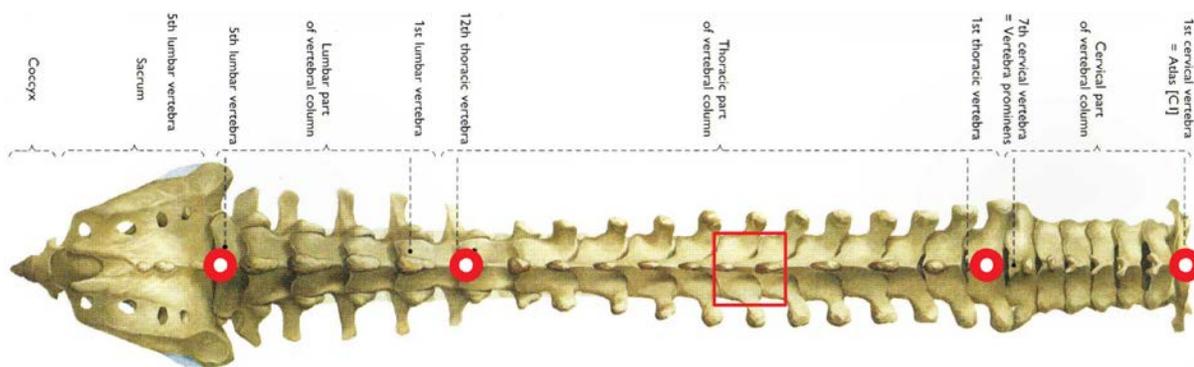


Figure 3-2. Spine with light sensor positions noted.

The 3D printed cube holds four sensors total, and measures 35mm on each side. A standard red laser (spot diameter 2.54mm) is mounted to the control handle, which is attached to the load cell mounted on the end effector of the manipulator. The photoresistors are monitored with a USB-6009 (National Instruments, Austin, TX) set to a 250hz sampling rate. The photoresistors not mounted on the cube are shielded using four 30° 3D printed cones, angled at 45° from vertical. The cones over sensors located at the C1 and L4 vertebrae are facing the head and feet of the virtual patient, respectively. The cones over sensors at T1 and T12 are facing the user. The purpose of these cones is to enforce a variety of task space positions and orientations. The actual apparatus is shown in Figure 3-3. The surgical handle and laser diode are visible at the top of the figure.

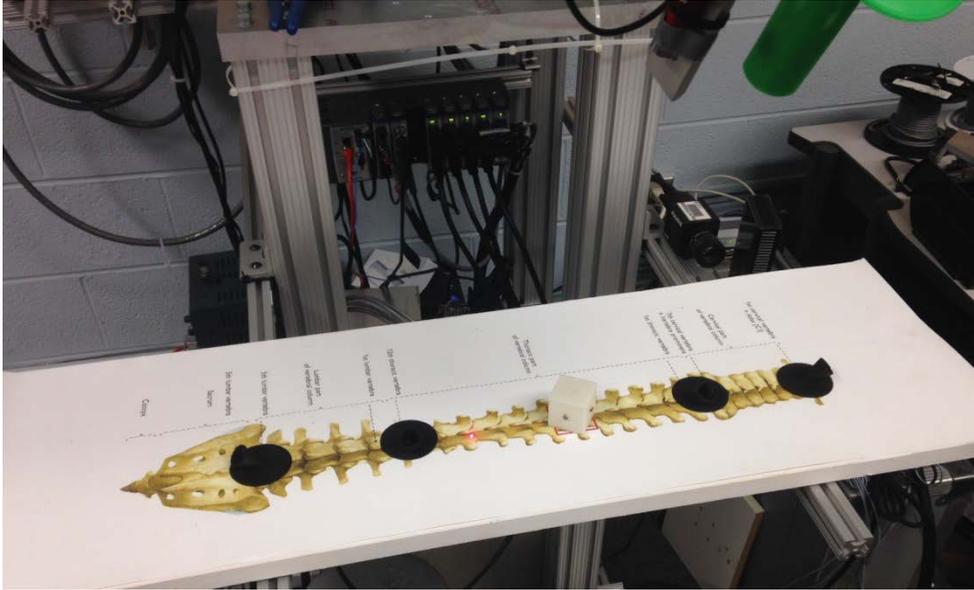


Figure 3-3. Actual apparatus with light sensors.

A group of 5 subjects (4 male) volunteered for the study. The task was to position the robotic manipulator so that the laser diode illuminated the photo sensor. A threshold of 1.5v was set, indicating an approximately 50% overlap between the photo sensor and laser spot. This threshold had to be held for 1s, at which point an audible beep was played, indicating to the subject to proceed to the next position. The subject was instructed to start with the sensor at C1, and proceed down the spine, visiting the sensors mounted on the cube in a clockwise fashion. After triggering the sensor at L5, the subject was instructed to return to the sensor at C1, stopping the trial. The trial was repeated 5 times per subject, and joint space, load cell forces/torques, and photosensor voltages were recorded through time. This dataset was collected for both the low controller gain ( $K = [8 \ 8 \ 8 \ 12 \ 12 \ 12]'$ ) and high controller gain ( $K = [12 \ 12 \ 12 \ 15 \ 15 \ 15]'$ ) settings. Time to complete the task, RMS force magnitude, and peak force magnitude are reported for each subject and each controller.

## Protection of Human Subjects

During haptic evaluation, the protection of human subjects is paramount. Therefore, there is a need for constant real time evaluation of safe operation. The primary danger exists when the robot must operate near the patient or the mounting hardware. The former presents obvious danger to the patient, and the latter presents danger to the surgeon in the form of a pinch injury. However, the patient can be represented by a bounding cylinder and the geometry of the mounting hardware is known, so countermeasures can be taken.

It is proposed that a so-called “danger-index” be used to quantify the risk of collision [37]. Logically, this index should consider the distance between the robot and the object to be avoided. As the safety index decreases, the maximum allowable task space velocity and control gains are reduced. This safety index is defined in Equation 3.9, where  $d_i$  is the distance between the end effector and object to be avoided, and  $b_i$  is the distance at which control gains should decrease. As the end effector position approaches the object to be avoided,  $x_i$  becomes less than  $b_i$ , resulting in a safety index less than 1. By multiplying the desired control gains against the safety index before applying them in the control law of Equation 3.7, the amount of torque provided by the joint motors for motion will decrease to zero as the object is approached, while gravity compensation torque remains unaffected.

$$S_i = \tanh \left| \frac{d_i}{b_i} \right| \quad (3.9)$$

During manual control of the robot, a joint limit (of either robot) may be encountered. Simply put, this is the limit approached as the end-effector is driven towards a position not achievable by one of the robots. For example, as the robot is pulled in one direction, eventually it will run out of travel. As this limit is approached, limit index similar to the safety index can be created, and applied in a similar fashion. As a limit is encountered, increasing resistance will be

felt by the surgeon, but he/she can continue to drive the robot through the position. Optionally, the hyperbolic tangent function can be bounded, limiting or eliminating the manipulator’s ability to impart a counter force on the user. This is more natural, more tunable, and safer than virtual spring based impedances as the counter force is a function of the applied force, not solely the end effector position in space.

Furthermore, none of the proposed positions put the operator’s hand in any pinch point positions. Additionally, the operator has a deadman switch linked to the robot emergency stop. This is similar to past protocols for human protection [12].

### Results

The results of the autonomous (pre-planned) control are shown in Figure 3-4. A move of 150mm in task space was commanded. Table 3-1 shows the settling times for a typical move between imaging views.

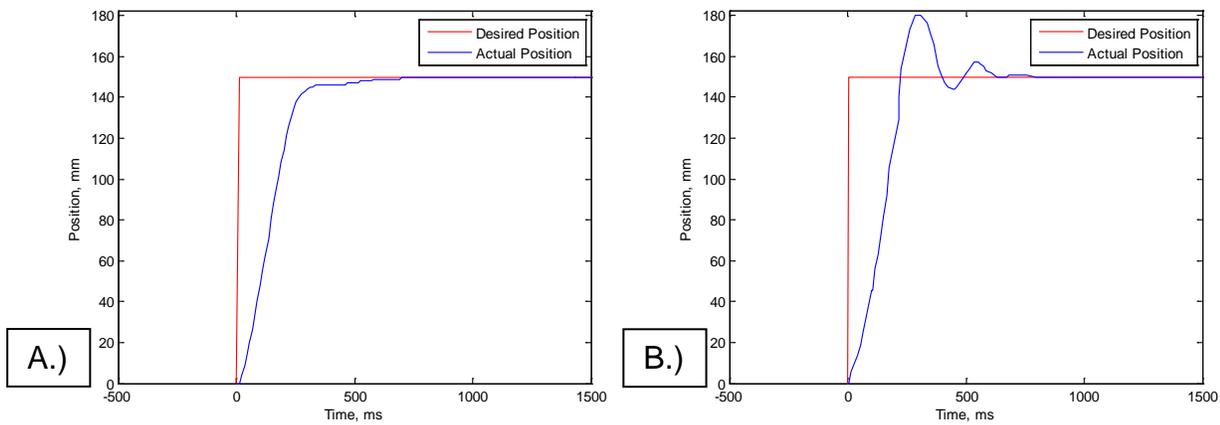


Figure 3-4. Controller performance for a typical 150mm move A.) PD/PI control B.) RISE control

Table 3-1. Settling times for a typical 150mm move

	P/PI Control	RISE Control
Settling time, e=1mm	567ms	629ms
Settling time, e=0mm	667ms	791ms

Figure 3-5 shows a typical collision avoidance scenario. In this scenario, the end effector of the robot is pushed towards a fixed object whose position is known. As the distance to the collision boundary approaches zero, the operator continues to push towards the boundary, noted by the increase in applied force at  $t=7$  seconds. The robot responds, increasing joint torques to compensate for the operator's applied force in the direction of the collision. This continues to 35N, at which point the operator relents. The robot does not move until the operator's applied force swings negative, pushing the end effector away from the collision boundary ( $t=8$  seconds). Table 3-2 shows the results of the nine position study. Time to completion, RMS and peak force values are averaged over the 5 trials to produce a single metric.

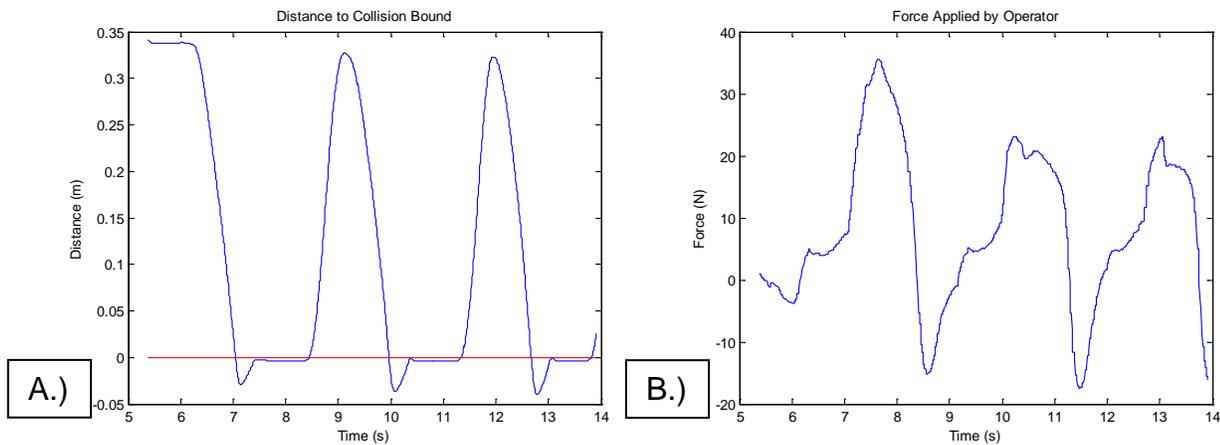


Figure 3-5. Controller performance for a near-collision A.) Distance to collision bound B.) Force applied by operator

Table 3-2. Time and force data for the nine position study

	Low Gains (mean values)			High Gains (mean values)		
	Time (s)	RMS (N)	Peak (N)	Time (s)	RMS (N)	Peak (N)
Subject 1	47.358	15.2	58.468	47.11	11.064	39.09
Subject 2	46.332	9.978	39.534	40.882	8.318	30.314
Subject 3	41.95	9.164	37.542	43.196	9.726	40.06
Subject 4	55.366	8.696	40.544	51.466	7.57	36.076
Subject 5	48.04	8.188	36.74	55.128	7.692	32.404

## **Discussion**

For the preplanned control task, the instantaneous response is primarily limited by the available control effort, as expected. It became apparent during the implementation of these controllers that the RISE controller, while stable, is difficult to tune. One set of gains may yield good (critically damped) performance, but that same set of gains in a different portion of the workspace may yield overshoot. Not only is the performance of the PD/PI control smoother, but it also reaches 1mm tracking error and 0mm tracking error faster. In fact, it is able to reach steady state in just under two-thirds of a second.

The collision avoidance strategy proved robust, even for fast approach velocities (0.5m/s) and small boundary thicknesses (2cm). The manipulator is able to present high stiffness and high counter force (>35N) at the collision boundary, while only requiring 5-7N of applied force for most control tasks in the workspace. It is able to stop the collision only 28 milliseconds after the collision boundary has been crossed, primarily as a result of the hyperbolic tangent function increasing smoothly as the boundary is crossed, instead of in a discontinuous fashion.

A framework was created to analyze the control system's performance in high precision positioning tasks using closed loop force feedback. The study required positioning of the center of the imaging plane within a window 2.5mm in diameter, while maintaining prescribed approach angles within +/-15 degrees. In both high and low gain cases, it took approximately 5 seconds on average to move the manipulator to the desired position and orientation, indicating smoothness and sufficient damping in the system to avoid hunting and overshoot that is often experienced in haptic control of heavy, high friction manipulators.

## **Conclusion**

Effective control schemes for the Mitsubishi PA10-6CE manipulator as a part of a surgical imaging platform have been demonstrated. In order to meet the requirements of the

platform, the manipulator must be able to complete both preplanned trajectories and function as haptic device. A traditional PD/PI scheme was found sufficient for the former, and explicit force control with a hyperbolic tangent based collision avoidance scheme was found suitable for the latter.

A potential improvement on the scheme described is direct compensation for the manipulator dynamics, such as the inertia and joint frictions. Such parameters have been experimentally measured for the PA10 and other manipulators [11], and could be implemented on faster real time processors, such as National Instruments' cRio 9081RT. The collision avoidance routine would also benefit from a faster real-time processor. Though lightweight algorithms exist such as the Lin-Canny closest features algorithm [38], collision detection was limited to direct calculation of the end-effector distance to the a plane in the workspace, in order to keep control loops above 500hz on the cRio 9024 processor.

## CHAPTER 4 TRAJECTORY PLANNING

In order to perform 3D reconstruction, it is necessary to collect a series of 2D images, or views, of the subject. These views may be snapshots as the “camera” travels along an arc path, or may be a more random, disorganized set of views. In the past, it was necessary to collect all of these views along a single semi-circular arc. However recent advancements in reconstruction algorithms have allowed for the contribution of off-axis views [8].

### **The Goal**

*The goal of this section is to develop and test novel path-planning algorithms for the collection of 2D images, for the case of stationary imaging (zero velocity while exposing).*

### **Description of Reconstruction Algorithm**

Dose reduction is always a goal in medical imaging, and one of the ways to reduce dose is to reduce the number of images needed for the reconstruction. Recent advances in reconstruction techniques have allowed for reduced radiological dose given the ability to perform non-circular imaging trajectories [8]. Given a rough 3-D shape, there are algorithms to identify the salient views, reducing the number of views necessary. For example, if the object to be imaged were a cube, it would be desirable to collect 3 or 6 images, each with a viewing axis perpendicular to the face of the cube. By doing this, all of the object’s geometry is collected with a minimum of images taken.

In addition to dose concerns, a greater number of images results in a longer imaging time, increasing wait and lengthening medical procedures. A generalized model for the object to be scanned exists given some information about the patient, and once the patient/object combination has been scanned once, subsequent scans will benefit further from the more precise model. Then, via gradient filtering, the models’ edge points can be detected. Running a Hough Transform[39]

on these edge points will create clusters of the radial distance and angular orientation of the more prominent edges of the object, which are most important. These clusters can be used to drive advantageous viewing angles (image plane rotation angles) which then become desired robot end-effectors positions and orientations. These positions and orientations are encoded as transformation matrices.

### **Problem Description**

Unfortunately, these desirable transformation matrices are unlikely to happen to be specified in an order advantageous to robot imaging, with respect to imaging time and mechanical velocity/acceleration constraints. Assuming reasonable velocities and accelerations, it may take up to 5 seconds to move between imaging positions. With ~30 images necessary to reconstruct a typical spine, total imaging time would be in excess of 2 minutes. This is not ideal – some procedures require multiple sets of reconstructions, and waiting 2-3 minutes for each would not be clinically desirable. However, the reconstruction algorithm does not depend upon the order of image collection, nor does any other part of the procedure.

Thus, motivation exists to sort these sets of end effector positions in order to minimize imaging time given some set of dynamic constraints. These constraints are primarily mechanical and nature, and they could be changed with the use of more powerful joint motors.

The problem of arranging a set of positions/orientations is a well-known problem, classified generally as the traveling salesman problem (TSP) [40]. The TSP is a NP-hard, NP-complete computational problem, [41][42] specifically a combinatorial optimization problem. It has been studied since at least the 1930s [43], and less formally prior to that. Salesmen must travel to a set of cities, but rarely care about the order of visitation. The path length can be minimized, but is subject to the following number of possible permutations, as long as symmetry is assumed.

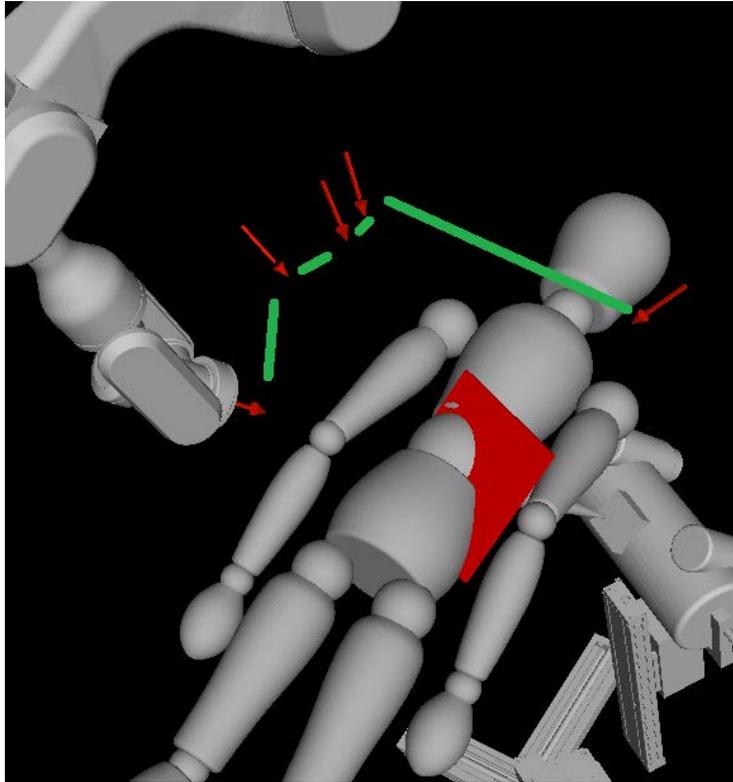


Figure 4-1. An example trajectory. Pictured are the PA-10 (above), patient, and UTR (below). The image plane is depicted in red, and the desired views with red arrows. The shortest trajectory is shown with green lines.

$$N_{paths} = \frac{(N_{views} - 1)!}{2} \quad (4.1)$$

Symmetry is defined as indifference in cost with regard to path direction (point A to B vs. B to A). This means that the number of possible paths increases dramatically as the number of points, or image views in this case, increase. For 20 views, there are  $6.08 \times 10^{16}$  views available. For 30 views,  $4.42 \times 10^{30}$  paths exist. In addition, the Mitsubishi PA-10-6CE robot has a maximum of 8 possible joint configurations for some end-effector position-orientations, and the under-table robot (UTR) has 2 possible joint configurations for some end-effector position-orientations.

Thus, the number of joint-space paths considering all robot inverse kinematic solutions is as follows:

$$N_{paths} = \frac{(N_{views} - 1)!}{2} \cdot 8^{N_{views}} \cdot 2^{N_{views}} \quad (4.2)$$

After considering the robots proposed, for 20 views,  $5.6 \cdot 10^{37}$  views exist. The possibility for multiple kinematic solutions from each robot only worsens the numerical problem that lies ahead. It is not computationally possible to evaluate all  $5.6 \cdot 10^{37}$  paths that exist in this case, within a reasonable period of time on current workstation hardware.

However, it is possible to compute all paths and analyze them for the shortest path for cases of approximately  $N < 10$ . Such an approach, commonly known as the brute force method [44], will be referred to as the Global Optimal Solution, or GOS. The GOS is one algorithm that guarantees finding the best path. Results for this are shown in Table 4-1.

Table 4-1. Time and path length for brute force

# of Views	# of Paths	Computational Time
5	120	0.049 seconds
9	362880	0.855 seconds
10	3628800	8.35 seconds
11	39916800	90.958 seconds
12	479001600	18.191 minutes
13	6227020800	3.94 hours
14	87178291200	2.29 days
15	1307674368000	34.48 days

There are dozens of alternative algorithms available that attempt to find the best path, but likely will result in a path that is marginally longer but can be found computationally in a reasonable amount of time. For example, the nearest neighbor algorithm (NN) can be used but this typically results in a path 25% longer than optimal [45], but can result in the worst possible tour for some cost matrices. Another alternative, Christofides algorithm, guarantees a solution

with path cost less than 150% of optimal [46]. Genetic algorithms have also been used with success in applications with multiple robot arms [47].

### Branch and Bound Algorithm

One of the inefficiencies associated with the brute force method is that solution trees are fully explored, even when it is known that their cost exceeds that of a previously fully populated path. An example is shown in Figure 4-2. The investigator has written an adaptation of the well-known Branch and Bound algorithm [48], [49].

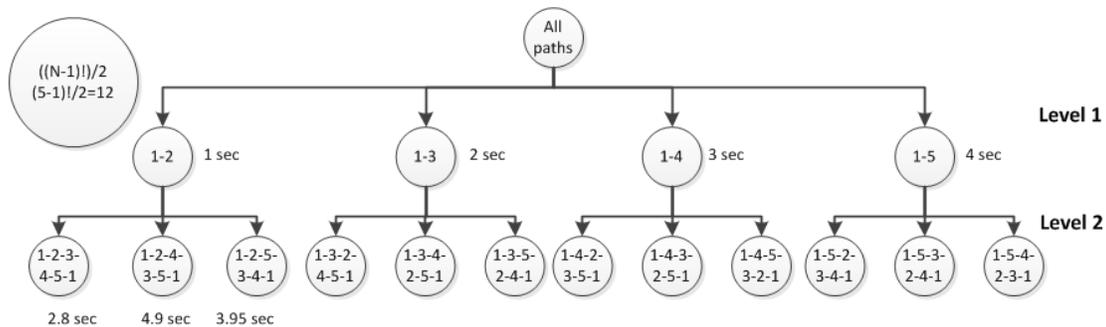


Figure 4-2. An Example Tree

Consider the example shown in Figure 4-2. By evaluating the solution tree down to the 1-2-3-4-5-1 solution, we can generate an upper bound of 2.8 seconds. Thus, trees containing subpaths from views 1-4 or 1-5 cannot contain optimal solutions, as the time cost to travel from point 1-4 or 1-5 already exceeds 2.8 seconds. This example is somewhat nonsensical, as the time cost from 1-4 or 1-5 cannot exceed the time cost from 1-2-3-4-5. A nonsensical example was used due to space limitations. The general principle holds that some trees will quickly have a cost greater than an established upper bound. Thus, the rest of the tree should be pruned, and left unexplored. In this case, half of the tree would be pruned, resulting in significant speed increases. Using the same CPU and programming language as before, the branch and bound algorithm can find the optimal path for 90 views in roughly the same time as the brute force

method took to find the optimal path for 9 views. Psuedo-code for the branch and bound scheme used is shown in Figure 4-3.

```
Reduce cost matrix C (row and column reduction)
While C is larger than 2x2
    Create matrix T
    For each C(i,j)=0, let T(i,j)=max(min(row Ci),min(col Cj))
    Choose (k,l) to maximize (T(k,l)). //the path without (k,l)
    Make a branch from current pair to pair ~(k,l).
    Let the cost be cost+T(k,l)
    Make a branch from current pair to pair (k,l) //path with(k,l)
    Delete row k and column l in C.
    Check for existing stored paths, find start and end cities m,n
    Set C(m,n) to infinity
    Reduce C, let s be sum of reducing constants
    Let the cost for path with k,l be cost + s
End
```

Figure 4-3 Branch and bound pseudo-code

The total computational time of the branch and bound routine for 6 sets of views ranging from  $n=30$  to  $n=200$  are shown in Table 4-2. For both the pseudo code in Figure 4-3 and the actual code above, the branch and bound approach is modified with the to-the-right variation, which states that the code exits when the last node to the right is evaluated, not when the guaranteed optimal solution is proven. When branching, the branch with the pair most likely to be included in the globally optimal path is selected. This is the path with the pair  $(k,l)$ , as shown in Figure 4-3. While it is likely that this pair is in the globally optimal solution, that is not guaranteed to be the case. To guarantee optimality, all branches with cost greater than the to-the-right cost much be explored until their cost exceeds the bound set in the to-the-right case.

Branch and bound can even be modified further to become branch and cut, which was utilized in 2005 to solve a problem visiting 33,810 points [50], and it was proven that no shorter tour exists. The computation took approximately 15.7 CPU-years to complete.

Table 4-2. Time and number of unique paths for branch and bound

n Views	# of paths	Computational Time
30	2.65253E+32	0.006 seconds
50	3.04141E+64	0.037 seconds
80	7.1569E+118	0.405 seconds
100	9.3326E+157	1.663 seconds
150	5.7134E+262	13.351 seconds
200	7.8865E+374	68.479 seconds

The computational time savings realized with the branch and bound algorithm increase further as the number of desired views increase – the branch and bound routine can calculate the best path for 200 views in the same time that the brute force method takes to calculate the best path for 11 views. For example, let us sort 20 random views, the generation of which is described in the methods section of Chapter 2. We have identified that the dynamics of the UTR are the primary limitation, as it is moving a payload 10 times greater in mass than the PA-10. Therefore, we calculate the desired end effector position of the UTR for each view, apply inverse kinematics, and use the joint-space positions of the UTR for the following example. Only 20 positions are used, to keep Figure 4-3 readable. We feed these views into the Branch and Bound program, with joint dynamics set to the parameters in Table 4-3. These dynamics are selected such that 50% of the available motor torque is never exceeded, to ensure that the stepper motors do not miss steps.

Table 4-3. The UTR’s dynamic constraints for path planning purposes

Joint	Velocity Limit	Acceleration Limit
1	150 mm/sec	500 mm/sec <sup>2</sup>
2	150 mm/sec	500 mm/sec <sup>2</sup>
3	45 °/sec	135 °/sec <sup>2</sup>
4	100 mm/sec	100 mm/sec <sup>2</sup>
5	30 °/sec	100 °/sec <sup>2</sup>

With the dynamic constraints set, a cost matrix  $C$  is calculated. The cells of  $C$  represent the cost to move from view  $i$  to view  $j$ . When  $i=j$ , the value is set to 99999, to denote a prohibited move, since we do not desire to “move” from the current view to the current view. Cost matrix  $C$  is shown in Figure 4-3, in units of seconds. After the cost matrix has been calculated, the branch and bound routine can begin.

$$C = \begin{bmatrix} 99999 & 1.941 & 1.725 & 1.288 & 2.173 & 1.983 & 1.124 & 1.508 & 1.930 & 2.813 \\ 1.941 & 99999 & 2.484 & 1.774 & 0.963 & 1.068 & 1.865 & 1.773 & 2.225 & 2.041 \\ 1.725 & 2.484 & 99999 & 2.668 & 2.201 & 3.219 & 1.055 & 1.044 & 1.185 & 4.192 \\ 1.288 & 1.774 & 2.668 & 99999 & 2.018 & 1.819 & 2.049 & 1.957 & 2.409 & 1.858 \\ 2.173 & 0.963 & 2.201 & 2.018 & 99999 & 1.352 & 1.866 & 1.556 & 1.942 & 2.325 \\ 1.983 & 1.068 & 3.219 & 1.819 & 1.352 & 99999 & 2.601 & 2.508 & 2.960 & 2.036 \\ 1.124 & 1.865 & 1.055 & 2.049 & 1.866 & 2.601 & 99999 & 1.031 & 1.586 & 3.574 \\ 1.508 & 1.773 & 1.044 & 1.957 & 1.556 & 2.508 & 1.031 & 99999 & 1.206 & 3.481 \\ 1.930 & 2.225 & 1.185 & 2.409 & 1.942 & 2.960 & 1.586 & 1.206 & 99999 & 3.933 \\ 2.813 & 2.041 & 4.192 & 1.858 & 2.325 & 2.036 & 3.574 & 3.481 & 3.933 & 99999 \end{bmatrix}$$

Figure 4-3. First 10 rows and columns of the cost matrix  $C$

The branch and bound routine starts by splitting all possible solutions into two groups. In this case, those groups are the paths containing the view pair (10,12), and the paths not containing the view pair (10,12). The *minimum* cost of each group is calculated and assigned to that node on the tree. The results of the first few branches are shown in Figure 4-4. Note that the second branching results in a path cost greater than the minimum path cost for the first branch’s non-selected node. This means the optimal solution *could* lie in that branch (i.e. the path *not* containing pair (10,12), but this is unlikely. Therefore, the branch to the right approach does not guarantee optimality, but it does result in consistent computational time required for a given  $N$  views. The lowest cost associated with an unevaluated branch (in this case, 22.28) serves as the lower bound for the global optimal solution. Since the to-the-right algorithm results in a path cost

of 25.11 (Table 4-4), we accept that there is little to be gained by exploring other branches (in this case, less than 3 seconds of imaging time).

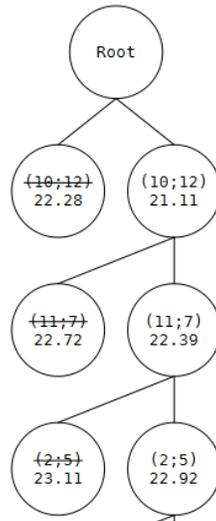


Figure 4-4. The first few nodes of the solution tree.

After completion of the branch and bound algorithm, we evaluate the total time it would take the UTR to traverse the path, visiting all of the views in the randomly sorted case and the optimally sorted case. The results are shown in Table 4-4.

Table 4-4. Time to complete both the random and optimally sorted paths

	Path Time (sec)
Random	56.18
Optimized	25.11

Clearly, the optimal solution saves imaging time, in this case a reduction of more than 50% of the unsorted path. The reasons become clear when we observe the joint positions in the sorted and unsorted cases. In Figure 4-4, Joint 3's trajectory over the 20 imaging views is shown.

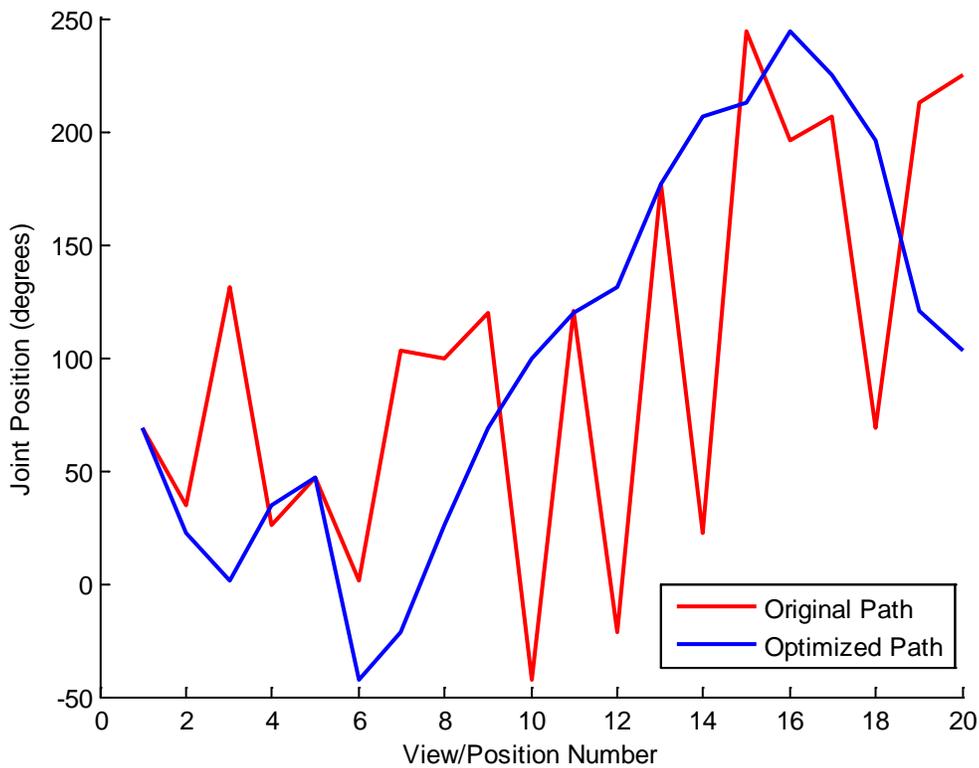


Figure 4-4. Unsorted and optimally sorted trajectories for Joint 3.

It has been demonstrated that solving the optimal path planning problem is important even for small view sets ( $n=18$ ). However, the problem becomes even more important to solve for more typical sets of desired images ( $n=30-50$ ). Imaging time savings on the order of 3-5 minutes can be realized, simply by sorting the views optimally.

### Conclusion

Although this research focuses primarily on typical imaging trajectories performed during spine surgery, there is interest in expanding the development to include other parts of the body. This path planning research is just as applicable to imaging other parts of the body, or any object in general. The results demonstrated present a compelling case for the use of a path planning optimization algorithm. For sets of views with roughly 60 views or less, the branch and bound algorithm can find globally optimal solutions in a few seconds of computational time. For larger

view sets, the branch and bound algorithm can be modified with the so-called to-the-right variation, which does not guarantee optimality but reaches a near optimal solution in a minute or less of computational time for sets of views 200 or less.

## CHAPTER 5 CT RECONSTRUCTION SENSITIVITY ANALYSIS

It is necessary for a positioner error sensitivity analysis to be performed for the reconstruction algorithm. Currently, no such analysis has been performed, so it is unknown if the inherent positional errors as a result of each robotic manipulator will be of consequence.

### **The Goal**

*A sensitivity study will be performed to examine the impact of manipulator inaccuracy on 3D reconstruction quality.*

### **Introduction**

The FDK algorithm [51], developed by research staff at Ford Motor Company in the early 1980s, was truly groundbreaking for the field of computed tomography. This algorithm allowed for direct three-dimensional reconstruction from two-dimensional projections, including projections produced from cone-beam computed tomography (CBCT). Because this algorithm is so prevalent, it is used as the basis of the sensitivity study. Many advancements have been made in the area of CBCT reconstruction, including iterative methods such as algebraic reconstruction methods, including the Simultaneous Algebraic Reconstruction Technique (SART) [52]. One of the advantages of SART is its robustness to low numbers of projections [53]. For years, the advantages of algebraic reconstruction techniques remained underutilized due to long reconstruction times. In the last 15 years, progress has been made in accelerating these reconstructions through commonly available consumer Graphics Processing Units (GPU) [54]. As the majority of portable CBCT machines to date use hardware fixed to a track, most error sensitivity research has been performed on the effects of known, repeatable positioning errors, typically due to mechanical sag [13]. Studies have shown that positioning errors with magnitude less than 2mm can result in measurable differences in quality of reconstruction, both subjectively

and objectively [14]. This study will examine the effects of unknown, non-measurable positioning error with magnitudes on the order of those reported for the Mitsubishi PA-10-6CE [11] and the UTR, the magnitudes of which are 1mm or less.

The FDK algorithm will be tested for sensitivity to positional error. This is error in actual position, when actual position is measured to be identical to desired, but the manipulator was not actually in that position. Such error is caused by encoder resolution limits and link/joint stiffness. By necessity, the backprojection algorithm will be supplied with images that were generated using position data that is not the same as the position data supplied to the forward projection algorithm. Therefore, it is necessary to study the effects of these errors in order to quantify their impact on reconstruction quality.

## Methods

A framework for the forward and back projection of existing voxelized volumes has been created<sup>1</sup>. This framework performs forward projection and backprojection using the CUDA (Compute Unified Device Architecture) parallel computing platform. This platform runs on NVIDIA GPUs. The use of a GPU allows for parallel processing, which yields improvements in image reconstruction time by an order of magnitude or more [54]. The CUDA platform is available in C, C++, and Fortran, with third party wrappers available for other languages. The projection code was written in C++ for best performance, and a MATLAB script was written to feed data to the projection routines and analyze the resulting data.

A 32-bit  $512^3$  volume consumes 524MB, so a card with at least 768MB of onboard memory is required. For this analysis, a GeForce GTX 660Ti with 3GB of onboard memory was

---

<sup>1</sup> Framework created in partnership with Klaus Mueller and Eric Papenhausen from Stony Brook University.

used. This is a present-day intermediate level consumer GPU, retailing for USD \$199 in 2013. On a GTX660, a reconstruction of a  $512^3$  volume can be performed in 5 seconds or less.

The gold standard dataset is a  $512^3$  phantom created from a cranial CT scan. Forward cone-beam projection is performed using a virtual detector with element size  $0.388 \times 0.776$ mm, and a detector count of  $1014 \times 374$  in x and y, respectively. In order to preserve fidelity, the forward projection has 2048 steps. A total of 360 projections are taken, equally spaced about a circular trajectory. The projections are taken with the *actual* translation matrices defining the position of the x-ray source and detector. The actual translation matrices are defined as the desired translation matrix, plus the robot positioner error. Therefore, it is possible to inject 6-degree of freedom error into the projection positions and orientations of both the source and the sensor. Six cases are evaluated. The 6 cases are described in Table 5-1.

Table 5-1. Error types for cases evaluated.

Case number	Description
1	no error
2	500 $\mu$ m, 0deg
3	1000 $\mu$ m, 0deg
4	500 $\mu$ m, 0.1deg
5	1000 $\mu$ m, 0.1deg
6	1800 $\mu$ m, 1.2deg

In the five cases for which error is added, the error is of a uniform distribution, added to the x, y, and z positions and orientation coordinates, with indicated magnitude. The error injected will vary from projection to projection, bounded by the magnitude listed, with uniform distribution. Cases 2 and 4 represent typical position and orientation errors for a calibrated Mitsubishi PA-10 robot [11], cases 3 and 5 represent twice the typical positioning errors of a calibrated PA-10, and case 6 represents the typical errors of an uncalibrated PA-10.

The projections are then input to the Feldkamp Davis Kress (FDK) filtered backprojection algorithm. During the backprojection, a ramp filter is used to reduce the blurring

inherent from the backprojection process. The output from the backprojection, a 3D reconstruction, will be analyzed mathematically using several standard metrics.

Image error was assessed by calculating the correlation coefficient, RMS volume error, and relative image error [53][55]. The 2-D correlation coefficient  $c = \text{corr}(f, f_0)$  was calculated using Equation 5.2:

$$r = \frac{\sum_i \sum_j (f_{ij} - \bar{f}_{ij})(f_{0ij} - \bar{f}_{0ij})}{\sqrt{(\sum_i \sum_j (f_{ij} - \bar{f}_{ij})^2)(\sum_i \sum_j (f_{0ij} - \bar{f}_{0ij})^2)}} \quad (5.2)$$

The center slice of the volume was selected for the 2-D correlation coefficient calculation. The RMS volume error was calculated using the following formula:

$$RMS_{error} = \frac{1}{IJK} \sqrt{\left( \sum_i \sum_j \sum_k (f_{ij} - f_{0ij})^2 \right)} \quad (5.3)$$

Finally, the relative error is calculated as follows:

$$e = \frac{\|f - f_0\|}{f_0} \quad (5.3)$$

In Equation 5.3,  $f$  is the image being evaluated, and  $f_0$  is the gold standard image. Both are of size 512x512 pixels.

Spatial resolution was assessed by calculating the spatial frequency response (SFR) or modulation transfer function (MTF) [56]. This is a sort of Bode plot for images, tracking image intensity as a function of image frequency. To perform the assessment, a sine wave phantom was created in MATLAB. One drawback to the sine wave approach to calculating the MTF is the impact of image noise on the measurement. Unfortunately, since the source volume is discretized, the use of a commonly known slanted-edge approach is not appropriate. Therefore, the sine wave virtual phantom was created with spatial frequency varying from 0.13 lp/mm to 1

lp/mm. After completing the reconstruction process, the MTF is calculated with Equation 5.1, using the center slice of the volume.

$$MTF(f) = \frac{\left( \frac{V(f)_{max} - V(f)_{min}}{V(f)_{max} + V(f)_{min}} \right)}{\left( \frac{V(0)_{max} - V(0)_{min}}{V(0)_{max} + V(0)_{min}} \right)} \quad (5.1)$$

In Equation 5.1,  $V(f)$  is the intensity of a single row of pixels through the sine wave phantom area.  $V(0)$  is the intensity in the low-frequency (DC) region – that is, the intensity of a pure black or pure white region. Note that since the imaging evaluation is done virtually, the virtual phantom has been discretized. The voxel size for the MTF evaluation is decreased to  $0.25\text{mm}^3$ , resulting in 4 available voxels for the highest spatial resolution tested, 1 lp/cm. This discretization means that at high spatial resolutions, there are not enough samples even in the original volume to fully sample the sine wave and capture its magnitude. As such, the MTF found by Equation 5.1 is divided by the MTF calculated from the original volume.

In addition to the clinical head CT phantom reconstruction, a reconstruction of the modified Shepp-Logan phantom is performed [57] [58]. We employ a 3D version, size  $512^3$ . The modification, described by Toft [58], is done to give better contrast. The reconstruction is performed with 360 equally spaced views, and injected errors as described in Table 5-1.

## Results

Table 5-2 presents the RMS error and correlation coefficient results for cases 1-5. The RMS and relative errors are quite small, owing primarily to the fact that the errors injected are unbiased. The results of the MTF evaluation are shown in Figure 5-1. Due to the aforementioned discretization issue, the results of the last point on the plot are subject to high error.

Table 5-2. RMS error and correlation coefficient for cases 1-5

	RMSe	e (%)	r
Case 1 - no error	0	0	1
Case 2 - 500um, 0deg	3.6997	0.392	0.999485
Case 3 - 1000um, 0deg	6.48125	0.798	0.998217
Case 4 - 500um, 0.1deg	4.82475	0.551	0.999077
Case 5 - 1000um, 0.1deg	6.95376	0.985	0.998001
Case 6 - 1800um, 1.2deg	18.7668	5.412	0.980662

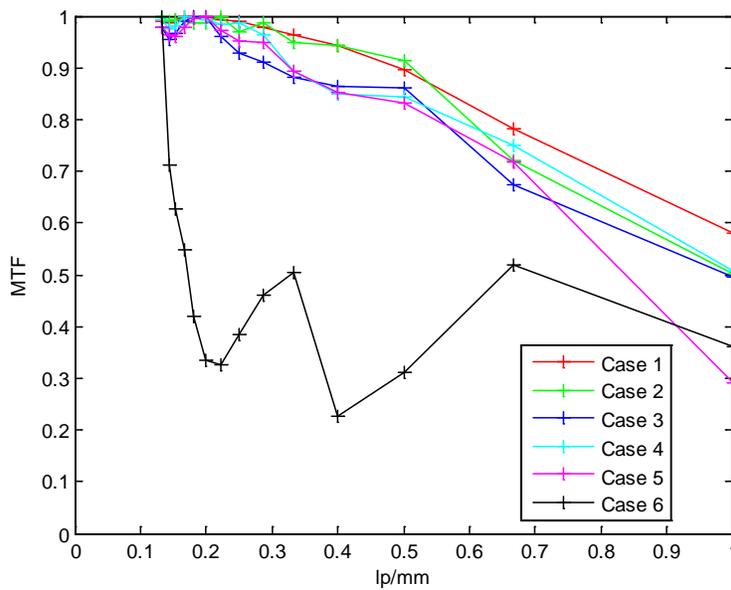


Figure 5-1. MTF as a function of spatial frequency for cases 1-6

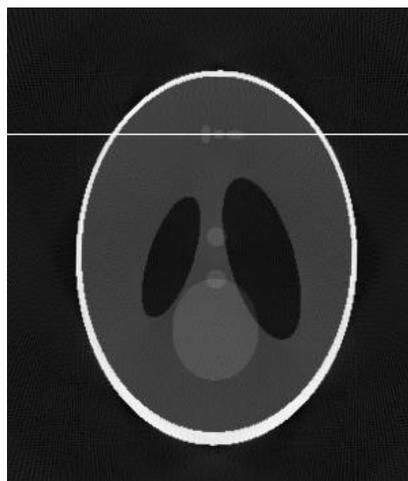


Figure 5-2. Modified Shepp-Logan phantom with white horizontal line indicating location of 1-D intensity plot

The intensity plots of a slice through the modified Shepp-Logan phantom are shown in Figure 5-3. The location of this 1-D slice is shown in Figure 5-3. The results of the head phantom are shown in Figure 5-4. The images are from the 256<sup>th</sup> slice of the volume, cropped and scaled to fit the figure. The arrow on case A (no error) and case E (twice the calibrated robot error) point out visible differences in image sharpness.

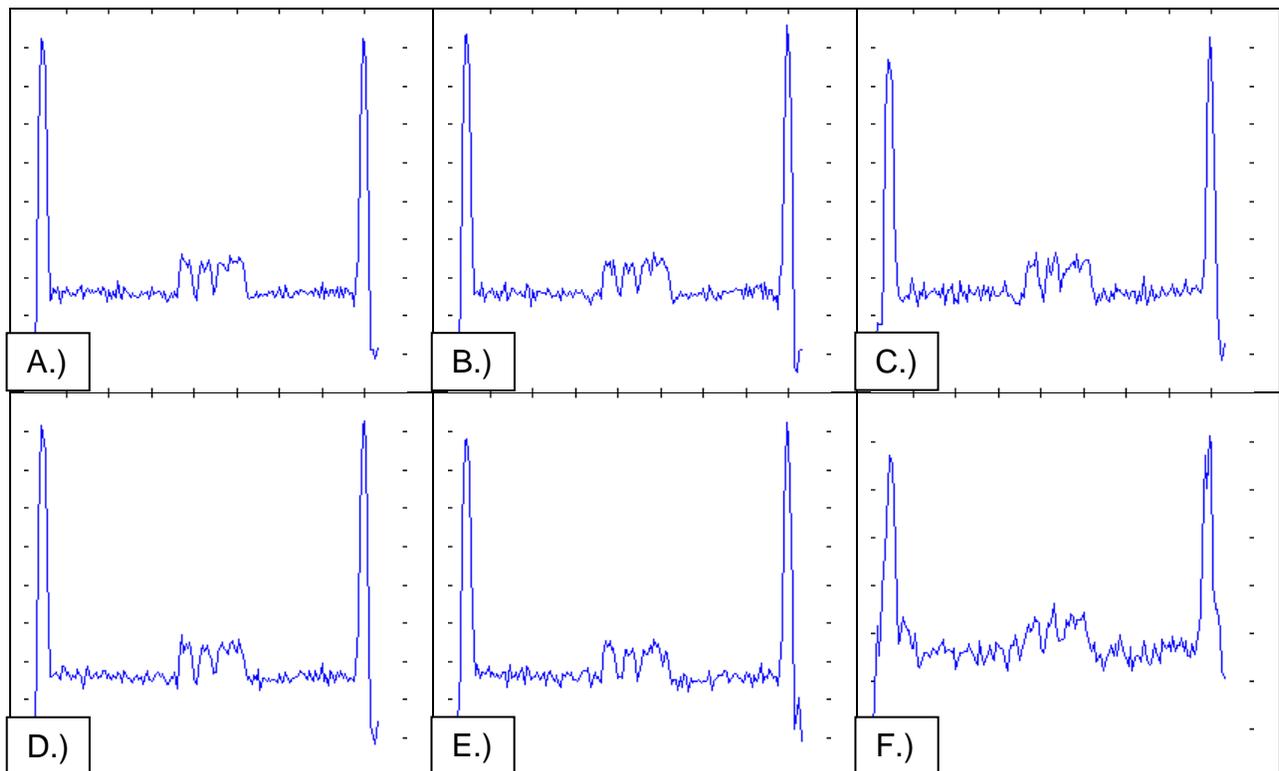


Figure 5-3. Intensity plots for a cross section of the modified Shepp-Logan phantom reconstruction A.) Case 1 B.) Case 2 C.) Case 3 D.) Case 4 E.) Case 5 F.) Case 6

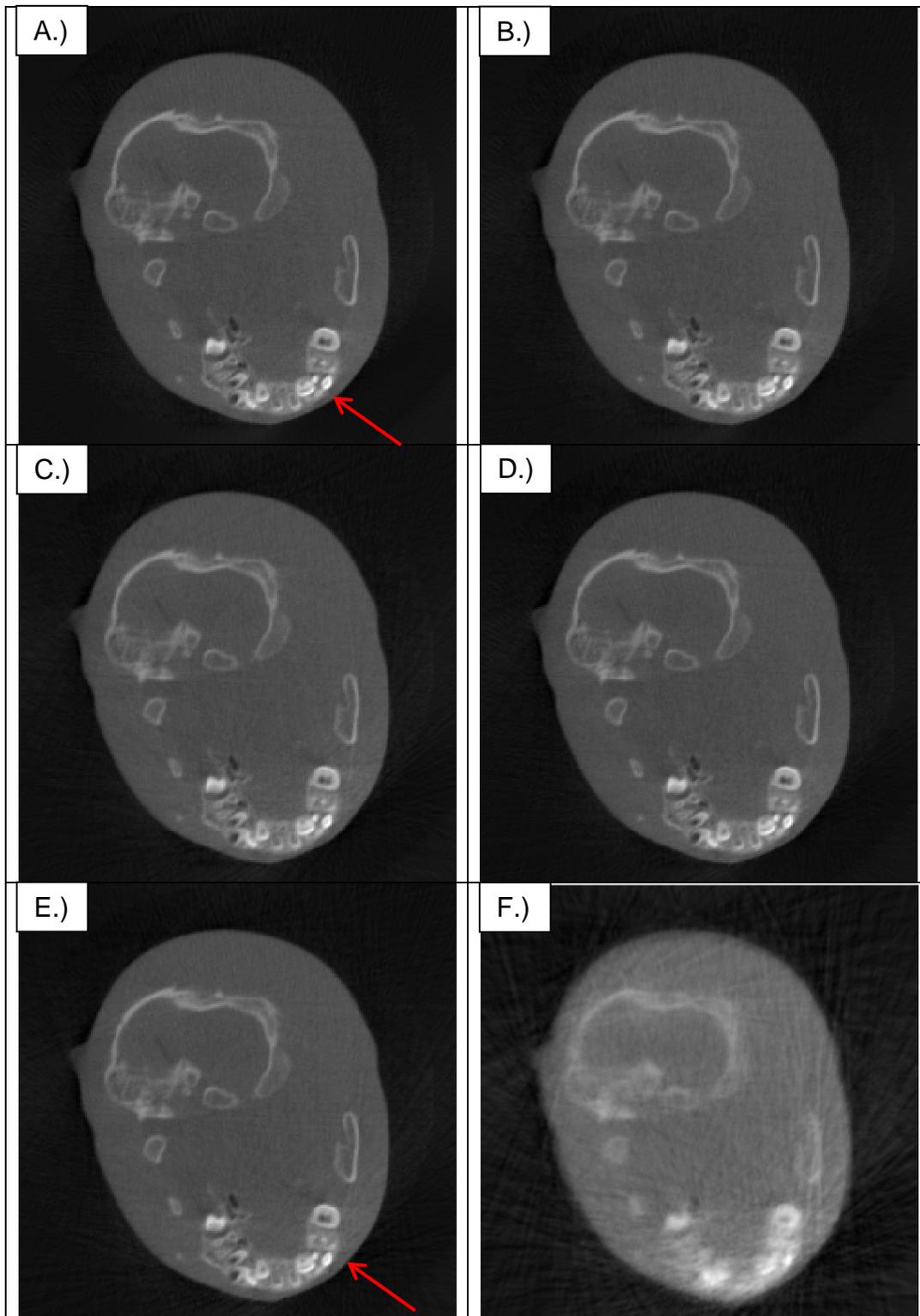


Figure 5-4. Reconstructions of cranial CT scan A.) Case 1 B.) Case 2 C.) Case 3 D.) Case 4 E.) Case 5 F.) Case 6

## Discussion

In order to generate design goals for manipulator-based imaging platforms, a study of the impact of positioning and orientation errors on reconstruction quality was necessary. The results of this study are promising. Very small image quality reductions are present in reconstructions performed with typical calibrated manipulator errors as compared to reconstructions without any error.

One drawback to the methodology performed here is that the phantom is a virtual phantom, discretized with voxels  $250\mu\text{m}$  per side. Therefore, the maximum spatial resolution that can be evaluated with a sine wave pattern virtual phantom is 1 lp/mm.

A review of other studies gives some insight into the fact that reconstruction quality degrades with positioning, but typically these studies are a result of the types of inaccuracies seen in clinical imaging platforms such as the C-arm. These inaccuracies are different in magnitude and uniformity as compared to the inaccuracies encountered with calibrated robotic imaging platforms. For instance, Jaffray et al. reported in 2002 that the typical mechanical sag of a CBCT gantry is on the order of 2mm, and that a simple calibration process improved the imaging [14]. However, no metrics (image error, MTF, intensity plots) were provided for the pre-post calibration comparison, only slices from the reconstruction and image subtraction. In 2012, Wicklein et al. reported on a method to perform misalignment correction through a markerless online-calibration procedure [59]. This procedure is helpful to correct for misalignments, but the error magnitudes they utilized (0 to 3 BPM, split between translations and rotations) are not the type of errors seen with calibrated robotic manipulators. As demonstrated in this study, positioning error accounts for the majority of the reconstruction degradation.

As seen in Table 5-2, the RMS and relative errors between the no error case (case 1) and cases with error (cases 2-6) are quite small, particularly for the calibrated robot error cases (cases

2-5). The correlation coefficients for the four calibrated robot error cases are very close to 1. As we would expect, the metrics point to decreasing image quality as error increases. Of note is the fact that moving from  $\pm 500\mu\text{m}$  errors to  $\pm 1000\mu\text{m}$  errors results in a greater degradation in image quality than simply adding  $0.1^\circ$  of error to the  $\pm 500\mu\text{m}$  case. The results from the MTF analysis are also encouraging, especially since we expect this metric, one of image sharpness, to be affected most by the uniform random error. Compared to the no error case, there is only marginal reduction in image sharpness in cases 2-5. Through most of the spatial resolution bandwidth evaluated, the decrease in MTF was approximately 0.1, or 10% of the DC value. Examination of the cross sectional intensity plots yields a few observations. First, the background noise increases marginally with increasing injected error. The difference in noise floor is particularly noticeable in cases 3 and 5, the cases with the greatest position (not orientation) error. The width and intensity of the 3 ellipsoids is relatively stable through all cases. Finally, an analysis of the reconstructed head phantom shown in Figure 5-3 provides a means of subjectively evaluating the resulting image quality. As shown, the images are nearly indistinguishable except near areas of detail and contrast, as indicated by the red arrows. There is little to no additional streaking or noise present in cases 2-5 as compared to the control.

Case 6 was included in the analysis to demonstrate clearly the necessity of calibrating robotic manipulators for CBCT imaging tasks. Using nominal geometric parameters provided by the manufacturer, and failing to account for joint flexibility, will result in inaccurate robot positioning and significantly degraded reconstruction results.

### **Conclusion**

Based on the quantitative and qualitative image metrics evaluated, the reconstruction quality is degraded very little for errors magnitudes of  $500\mu\text{m}/0.1^\circ$  or less. For larger errors, the impact to reconstruction quality is primarily in the area of image sharpness, but is still difficult to

notice visually. Some of this image blurring can be removed through the choice of filter sharpness during the filtered backprojection stage. At higher injected error levels there are slightly increased streak artifacts as well, which may obscure the visibility of low-contrast objects. One thing to remember is that no imaging system is completely free of positioner error, including existing C-arm, O-arm, and other CBCT techniques. This means that the “gold standard” no error case discussed is not achievable in practice. Therefore, given that robotic manipulators of suitable reach and payload can be calibrated to within  $\pm 500\mu\text{m}$  [11], it has been demonstrated that CBCT imaging can be performed with separate robotic manipulators and no external motion capture hardware with almost negligible loss of image quality.

## CHAPTER 5 CONCLUSION

The surgeon interactive imaging system described here is the first of its kind. This system can provide reduced patient dose, improved image reconstruction, and better usability in surgery by adding degrees of freedom to the image plane, opening viewing angles previously unavailable. The key contributions are outlined as follows:

- Design, assembly, and accuracy validation of a novel 5-DoF robotic manipulator for positioning of the x-ray source
- Design and evaluation of a haptic control law for the surgeon-interactive portion of the imaging task
- A framework for optimally forming trajectories through the desired views
- A study of the sensitivity of FDK reconstruction algorithm to positioning errors associated with the robotic manipulators selected for use

The new platform shows feasibility not only for improved spine surgery imaging, but also expansion to perform gait and stairclimbing studies simply by swapping out one of the prismatic linear stages for a longer version. These stages are commonly found in variants over 10ft, allowing capture of an entire stride, both swing and stance phase. Overall, this work has potential to make significant contributions to the research area by providing a flexible platform for imaging studies, and eventually provide a practical replacement for existing O-arm and C-arm based solutions.

## APPENDIX BACK PROJECTION CONFIGURATION

### README FOR BACK PROJECTOR (beta)

---

#### CONFIG FILE INSTRUCTIONS:

"input file volume: " --File containing the projections to be backprojected (FDK)  
"input file angles: " --File containing a list of angles (a sample has been provided, see raw\_AngleN.txt)  
"output file: " --File location where the backprojected volume will be written to  
"number of projections: " --Number of projections to be backprojected  
"voxel size (isotropic): " --Isotropic voxel size  
"isocenter: " --Isocenter  
"detector element size x (mm): " --Horizontal size of the detector element in millimeters  
"detector element size y (mm): " --Vertical size of the detector element in millimeters (cone beam only)  
"number of detectors x: " --Number of detector elements in the trans-axial direction  
"number of detectors y: " --Number of detector elements in the axial direction  
"source to isocenter: " --Distance from the source to isocenter  
"isocenter to detector: " --Distance from the isocenter to detector  
"volume size: " --Number of cubic voxels

#### CONFIG FILE EXAMPLE

---

```
input file volume: projections.raw
input file angles: raw_AngleN.txt
output file: result.vol
number of projections: 364
voxel size (isotropic): 0.5
isocenter: -127.75
detector element size x (mm): 0.388
detector element size y (mm): 0.776
number of detectors x: 1014
number of detectors y: 374
source to isocenter: 647.7
isocenter to detector: 520.7
volume size: 512
```

---

This will perform a cone beam backprojection for 364 projections. The projections contain 1014 detector cells; each detector cell is 0.388mm long. The resulting volume is written to "result.vol" and can be viewed using the imageJ viewing tool.

#### HELPFUL LINKS

ImageJ Website: <http://rsbweb.nih.gov/ij/>

## INSTRUCTIONS FOR IMAGEJ VIEWING

go to:

File->Import->Raw (Select volume file)

Image type: 32-bit Real

Width: volume size (e.g. 512)

Height: volume size

Offset to first image: 0

Number of images: volume size

Gap between images: 0

Check "Little-endian byte order"

Select OK

## LIST OF REFERENCES

- [1] B. I. Martin, R. a Deyo, S. K. Mirza, J. a Turner, B. a Comstock, W. Hollingworth, and S. D. Sullivan, "Expenditures and health status among adults with back and neck problems.," *JAMA*, vol. 299, no. 6, pp. 656–64, Feb. 2008.
- [2] Y. Rampersaud, D. Simon, and K. Foley, "Accuracy requirements for image-guided spinal pedicle screw placement.," *Spine (Phila. Pa. 1976)*, vol. 26, no. 4, pp. 352–359, Feb. 2001.
- [3] U. Gille, "Vertebral Column Colored - Public Domain." [Online]. Available: [http://en.wikipedia.org/wiki/File:Gray\\_111\\_-\\_Vertebral\\_column-coloured.png](http://en.wikipedia.org/wiki/File:Gray_111_-_Vertebral_column-coloured.png).
- [4] Hatchmpi, "No Title." [Online]. Available: [http://www.hatchmpi.com/FLUOROSCOPIC\\_C-ARMS\\_main.htm](http://www.hatchmpi.com/FLUOROSCOPIC_C-ARMS_main.htm).
- [5] "Medical Imaging Equipment." [Online]. Available: <http://www.medimagingsales.com>.
- [6] R. Betz and P. Noegel, "Radiography device," *US Pat. 6,435,715*, 2002.
- [7] L. Mitsubishi Heavy Industries, "Instruction Manual for Installation, Maintenance, and Safety." Tokyo, Japan, 2003.
- [8] Z. Zheng and K. Mueller, "Identifying sets of favorable projections for few-view low-dose cone-beam CT scanning," *11th Int. Meet. Fully Three-Dimensional Image Reconstr. Radiol. Nucl. Med.*, pp. 314–317, 2011.
- [9] E. A. Pearson, S. Member, S. Cho, and C. A. P. Member, "Non-circular Cone Beam CT Trajectories : A Preliminary Investigation on a Clinical Scanner," vol. 0, no. 1, pp. 3172–3175, 2010.
- [10] K. Xu, F. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 3, pp. 654–663, Jun. 2005.
- [11] C. A. Lightcap, "Measurement and control issues in a novel dynamic radiographic imaging system," pp. 1–132, 2008.
- [12] J. D. Yamokoski, "Safe and Robust Sensor Motion Planning and Control," pp. 1–120, 2009.

- [13] J. H. Siewerdsen, D. J. Moseley, S. Burch, S. K. Bisland, a. Bogaards, B. C. Wilson, and D. a. Jaffray, "Volume CT with a flat-panel detector on a mobile, isocentric C-arm: Pre-clinical investigation in guidance of minimally invasive surgery," *Med. Phys.*, vol. 32, no. 1, p. 241, 2005.
- [14] D. a Jaffray, J. H. Siewerdsen, J. W. Wong, and A. a Martinez, "Flat-panel cone-beam computed tomography for image-guided radiation therapy.," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 53, no. 5, pp. 1337–49, Aug. 2002.
- [15] P. Automation, "402/403/404XE Series Product Manual." Cleveland, Ohio, 2012.
- [16] C. Mavroidis, S. Dubowsky, P. Drouet, J. Hintersteiner, and J. Flanz, "A systematic error analysis of robotic manipulators: application to a high performance medical robot," *Proc. Int. Conf. Robot. Autom.*, vol. 2, pp. 980–985, 1997.
- [17] J. Santolaria, J.-A. Yagüe, R. Jiménez, and J.-J. Aguilar, "Calibration-based thermal error model for articulated arm coordinate measuring machines," *Precis. Eng.*, vol. 33, no. 4, pp. 476–485, Oct. 2009.
- [18] Lipták, *Instrument engineers' handbook : process measurement*. Radnor, Pa: Chilton Book Co, 1982.
- [19] N. Instruments, "Stepper Motors and Encoders." [Online]. Available: <http://sine.ni.com/ds/app/doc/p/id/ds-311/lang/en>.
- [20] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.
- [21] B. Mooring, Z. S. Roth, and M. R. Driels, *Fundamentals of manipulator calibration*. J. Wiley, 1991.
- [22] D. T. Bui, Tanh M. Chow, "Stall Detection Circuit and Method," US Patent No. 69006572005.
- [23] J. Zambelli, B. E. Nett, S. Leng, C. Riddell, B. Belanger, and G.-H. Chen, "Novel C-arm based cone-beam CT using a source trajectory of two concentric arcs," p. 65101Q–65101Q–10, Mar. 2007.
- [24] D. Soimu and N. Pallikarakis, "RECONSTRUCTIONS USING FDK ALGORITHM," pp. 3–6.
- [25] I. J. Hill, *A Novel Testing Platform for Characterizing Cervical Spine Biomechanics*. Gainesville, Fla: University of Florida, 2013.
- [26] "DRICS - Dynamic Radiographic Imaging Control Software," 2013. [Online]. Available: <http://code.google.com/p/drics/>.

- [27] W. E. Snyder, *Industrial robots: computer interfacing and control*. Prentice-Hall, 1985.
- [28] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, “Asymptotic Tracking for Uncertain Dynamic Systems Via a Multilayer Neural Network Feedforward and RISE Feedback Control Structure,” *Autom. Control. IEEE Trans.*, vol. 53, no. 9, pp. 2180–2185, 2008.
- [29] R. Van der Linde and P. Lammertse, “The HapticMaster, a new high-performance haptic interface,” *Proc. ...*, pp. 1–5, 2002.
- [30] G. Zeng and A. Hemami, “An overview of robot force control An overview of robot force control Ganwen Zeng and Ahmad Hemami,” vol. 15, no. 05, pp. 473–482, 2013.
- [31] J. Wahrburg, “Improving robot arm control for safe and robust haptic cooperation in orthopaedic procedures,” no. October, pp. 316–322, 2007.
- [32] D. E. Whitney, “Historical Perspective and State of the Art in Robot Force Control,” *Int. J. Rob. Res.*, vol. 6, no. 1, pp. 3–14, Mar. 1987.
- [33] C. D. Crane and J. Duffy, *Kinematic Analysis of Robot Manipulators*. Cambridge University Press, 2008.
- [34] Danaher, “PACIFIC SCIENTIFIC T-SERIES NEMA 23 HIGH TORQUE MOTORS,” Washington, D.C.
- [35] R. Paul and B. Shimano, “Kinematic control equations for simple manipulators,” *Decis. Control Incl. 17th ...*, 1978.
- [36] S. C. Cannon and G. I. Zahalak, “The mechanical behavior of active human skeletal muscle in small oscillations,” *J. Biomech.*, vol. 15, no. 2, pp. 111–121, 1982.
- [37] D. Kulić and E. A. Croft, “Real-time safety for human–robot interaction,” *Rob. Auton. Syst.*, vol. 54, no. 1, pp. 1–12, 2006.
- [38] M. Lin and J. Canny, “A fast algorithm for incremental distance calculation,” *Robot. Autom. 1991. ...*, 1991.
- [39] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
- [40] M. M. Flood, “The Traveling-Salesman Problem,” *Oper. Res.*, vol. 4, no. 1, pp. pp. 61–75, 1956.
- [41] D. Johnson and L. McGeoch, “The traveling salesman problem: A case study in local optimization,” *Local search Comb. Optim.*, pp. 1–103, 1997.

- [42] J. Clausen, “Branch and bound algorithms-principles and examples,” *Dep. Comput. Sci. Univ. ...*, pp. 1–30, 1999.
- [43] V. K. Menger, A. Wunsch, and S. Probleme, “Bericht tiber ein mathematisches Kolloquium,” 1929.
- [44] J. Bentley, “No Title,” *Unix Review*, 1997.
- [45] G. Gutin, A. Yeo, and A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP,” *Discret. Appl. Math.*, no. January, 2002.
- [46] N. Christofides and C.-M. U. P. P. A. M. S. R. GROUP., *Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*. Defense Technical Information Center, 1976.
- [47] E. Xidias, “Time-optimal task scheduling for two robotic manipulators operating in a three-dimensional environment,” *Proc. ...*, vol. 224, no. 7, pp. 845–855, Nov. 2010.
- [48] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, “An Algorithm for the Traveling Salesman Problem,” *Oper. Res.*, vol. 11, no. 6, pp. pp. 972–989, 1963.
- [49] S. Dubowsky and T. D. Blubaugh, “Planning time-optimal robotic manipulator motions and work places for point-to-point tasks,” *Robot. Autom. IEEE Trans.*, vol. 5, no. 3, pp. 377–381, 1989.
- [50] D. L. Applegate, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.
- [51] L. Feldkamp, L. Davis, and J. Kress, “Practical cone-beam algorithm,” *JOSA A*, vol. 1, no. 6, pp. 612–619, 1984.
- [52] A. H. Andersen and A. C. Kak, “Simultaneous Algebraic Reconstruction Technique (SART): A superior implementation of the {ART} algorithm,” *Ultrason. Imaging*, vol. 6, no. 1, pp. 81–94, 1984.
- [53] W. Chlewicki, C. Badea, and N. Pallikarakis, “FOR LIMITED NUMBER OF PROJECTIONS,” pp. 3–5.
- [54] K. Mueller and R. Yagel, “On the use of graphics hardware to accelerate algebraic reconstruction methods,” in *SPIE Medical Imaging Conference*, 1999, vol. 3659, no. 2, pp. 615–625.
- [55] X. Jia, Y. Lou, J. Lewis, R. Li, X. Gu, C. Men, W. Y. Song, and S. B. Jiang, “GPU-based fast low-dose cone beam CT reconstruction via total variation.,” *J. Xray. Sci. Technol.*, vol. 19, no. 2, pp. 139–54, Jan. 2011.

- [56] P. Burns, “Slanted-edge MTF for digital camera and scanner analysis,” *IS TS PICS Conf.*, 2000.
- [57] L. A. Shepp and B. F. Logan, “The Fourier reconstruction of a head section,” *Nucl. Sci. IEEE Trans.*, vol. 21, no. 3, pp. 21–43, 1974.
- [58] P. Toft, “‘The Radon Transform, Theory and Implementation’ (unpublished dissertation).”
- [59] J. Wicklein, H. Kunze, W. a Kalender, and Y. Kyriakou, “Image features for misalignment correction in medical flat-detector CT.,” *Med. Phys.*, vol. 39, no. 8, pp. 4918–31, Aug. 2012.

## BIOGRAPHICAL SKETCH

Timothy Zane Elmore was born in Fort Lauderdale, Florida. He graduated from the University of Florida in 2008 with a B.S. in Mechanical Engineering, and a B.S. in Aerospace Engineering. He joined the Orthopaedic and Biomechanics Laboratory at the University of Florida in 2008 to begin research in robotics and medical engineering. He became a PhD candidate in the field of Mechanical Engineering in 2012, and graduated with a PhD in Mechanical Engineering from the University of Florida in 2013.